CA Access Control

Reference Guide

12.6.01

Third-Party Notices

CONTAINS IBM(R) 32-bit Runtime Environment for AIX(TM), Java(TM) 2 Technology Edition, Version 1.4 Modules

Sample Scripts and Sample SDK Code

The Sample Scripts and Sample SDK code included with the CA Access Control product are provided "as is", for informational purposes only. They may need to be adjusted in specific environments and should not be used in production without testing and validating them before deploying them on a production system.

CA Technologies does not provide support for these samples and cannot be responsible for any errors that these scripts may cause.

CA Technologies Product References

This document references the following CA Technologies products:

- CA Access Control Enterprise Edition
- CA Access Control
- CA Single Sign-On (CA SSO)
- CA Top Secret®
- CA ACF2™
- CA Audit
- CA Network and Systems Management (CA NSM, formerly Unicenter NSM and Unicenter TNG)
- CA Software Delivery (formerly Unicenter Software Delivery)
- CA Service Desk (formerly Unicenter Service Desk)
- CA User Activity Reporting Module (formerly CA Enterprise Log Manager)
- CA Identity Manager

Documentation Conventions

The CA Access Control documentation uses the following conventions:

Format	Meaning
`Mono-spaced font`	Code or program output
Italic	Emphasis or a new term
Bold	Text that you must type exactly as shown
A forward slash (/)	Platform independent directory separator used to describe UNIX and Windows paths

The documentation also uses the following special conventions when explaining command syntax and user input (in a mono-spaced font):

Format	Meaning
Italic	Information that you must supply
Between square brackets ([])	Optional operands

Format	Meaning
Between braces ({})	Set of mandatory operands
Choices separated by pipe (|).	Separates alternative operands (choose one). For example, the following means *either* a user name *or* a group name: {*username*|*groupname*}
...	Indicates that the preceding item or group of items can be repeated
Underline	Default values
A backslash at end of line preceded by a space (\)	Sometimes a command does not fit on a single line in this guide. In these cases, a space followed by a backslash (\) at the end of a line indicates that the command continues on the following line. **Note:** Avoid copying the backslash character and omit the line break. These are not part of the actual command syntax.

Example: Command Notation Conventions

The following code illustrates how command conventions are used in this guide:

```
ruler className [props({all|{propertyName1[,propertyName2]...})]
```

In this example:

- The command name (ruler) is shown in regular mono-spaced font as it must be typed as shown.
- The *className* option is in italic as it is a placeholder for a class name (for example, USER).
- You can run the command without the second part enclosed in square brackets, which signifies optional operands.
- When using the optional parameter (props), you can choose the keyword *all* or, specify one or more property names separated by a comma.

File Location Conventions

The CA Access Control documentation uses the following file location conventions:

- *ACInstallDir*—The default CA Access Control installation directory.
  - Windows—C:\Program Files\CA\AccessControl\
  - UNIX—/opt/CA/AccessControl/

- *ACSharedDir*—A default directory used by CA Access Control for UNIX.
  - UNIX—/opt/CA/AccessControlShared
- *ACServerInstallDir*—The default CA Access Control Enterprise Management installation directory.
  - /opt/CA/AccessControlServer
- *DistServerInstallDir*—The default Distribution Server installation directory.
  - /opt/CA/DistributionServer
- *JBoss_HOME*—The default JBoss installation directory.
  - /opt/jboss-4.2.3.GA

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Documentation Changes

The following documentation updates have been made since the last release of this documentation:

- Utilities— Updated chapter with updates to the following utilities, services, and daemons:
  - dmsmgr -sync function
  - seaudit Utility—Display Audit Log Records
  - secons -i Function
  - sepmd Utility—Administer Subscribers and the Update File
  - uxauth_selinux.sh—Enable SElinux Support
  - uxconsole Utility -map
  - uxconsole Utility -manage
  - uxconsole Utility -krb
  - uxconsole Utility -debug
  - uxpreinstall Utility
- Configuration Files— Updated chapter with the following new or changed sections:
  - accommon.ini File communication
  - accommon.ini File ReportAgent
  - accommon.ini File AccountManager
  - seos.ini Initialization File daemons
  - seos.ini Initialization File OS_User
  - seos.ini Initialization File passwd
  - seos.ini Initialization File policyfetcher (UNIX config section)
  - seos.ini Initialization File seos
  - seos.ini Initialization File seoswd
  - seos.ini Initialization File serevu
  - pmd.ini File endpoint_management
  - pmd.ini File passwd
  - audit.cfg File—Filter Audit Records
  - uxauth.ini File ad
  - uxauth.ini File agent

- uxauth.ini File register
- SSH Device XML File

- Registry Entries— Updated chapter with updates to the following entries:
  - DMS Endpoint Management
  - policyfetcher (Windows registry key)
  - ReportAgent Key—Registry Settings
  - Additional Registry Keys
- Audit Log Records- Updated chapter with updates to the following records:
  - Audit Event Types-Trace Message On a User
- Used Ports- Updated chapter with updates to the following ports:
  - UNIX Endpoint Used Ports
  - Windows Endpoint Used Ports
  - Server Components Used Ports
  - UNIX Authentication Broker Used Ports
  - Privileged User Password Management Used Ports
  - ObserveIT Used Ports
  - UARM Used Ports

Contents

Chapter 1: Introduction

This section contains the following topics:

About this Guide

This guide provides information about CA Access Control utilities, configuration files, status codes and messages, and so on. This guide is also provided with CA Access Control Enterprise Edition, which offers enterprise management and reporting capabilities, and advanced policy management features.

To simplify terminology, we refer to the product as CA Access Control throughout the guide.

Who Should Use this Guide

This guide was written for security and system administrators who are executing commands or maintaining and configuring a CA Access Control-protected environment.

Chapter 2: Utilities

CA Access Control has many utilities. As a convenient overview, this chapter describes them in alphabetical order.

acpwd Utility—Check In and Check Out Privileged Account Passwords

Use the Privileged User Password Management Agent to obtain privileged account password from the CA Access Control endpoint. Running the Privileged User Password Management Agent using a command line lets you connect to CA Access Control Enterprise Management and check out, check out, and retrieve privileged account passwords.

This command has the following format:

`acpwd {-checkin | -checkout | -get} -account` *`name`* `-ep` *`name`* `-eptype` *`type`* `-container` *`name`* `-[timeout <`*`timeout`*`>] [-nologo] [-help]`

-checkin

Executes the privileged account password check in process.

-checkout

Executes the privileged account password check out process.

-get

Retrieves the privileged account password without executing the check out process.

-account *name*

Defines the privileged account password to check out or check in.

-ep *name*

Defines the name of the endpoint where the privileged account resides.

-eptype *type*

Specifies the type of the endpoint.

Example: Windows Agentless

-container *name*

Defines the name of the container that the account resides on.

-nologo

Specifies that the output displays only the password without any additional information.

-timeout *timeout*

Specifies the timeout period, in seconds, to wait for a response from the server.

-help

Displays the help file.

acuxchkey Utility—Change Encryption Key Settings

Use the acuxchkey utility to change encryption key and Message Queue settings. This command has the following format:

```
acuxchkey -t -pwd password
```

-t

Specifies the Message Queue change option.

-pwd *password*

Defines the Message Queue password.

Example: Change Message Queue Password

This command saves the changed Message Queue encrypted password in the database. The password is "secret", and must be in clear text and enclosed in double quotes:

```
acuxchkey -t -pwd "secret"
```

Example: Change Distribution Server Communication Settings

This example shows you how to change the Distribution Server settings to work with SSL:

```
env config
editres CONFIG accommon.ini section (communication) token (Distribution_Server)
value ("ssl://DS_host:7243")
```

More information:

sechkey Utility—Change the Message Queue Password (see page 115)

ChangeEncryptionMethod Utility—Change Encryption Method

Valid on UNIX

The ChangeEncryptionMethod utility changes the encryption methods.

Note: This utility is supplied as a script file and is located in the lbin directory.

When you run this utility, you can choose one of the following encryption methods:

- DEFAULT
- AES (128bit, 192bit, or 256bit)
- DES
- TRIPLEDES
- SCRAMBLE

If you do not specify an encryption method, the utility prompts you for it. The utility searches for existing Policy Models in the system, decrypts them by running "sepmd -de pmd_name", and then changes the encryption method by linking libcrypt to the new shared library: libaes128, libaes192, libaes256, libdes, libtripledes, or libscramble.

Note: To run this utility, CA Access Control must be running. To change the encryption method, the script asks you whether it can temporarily shut down CA Access Control.

Important! Verify that you use identical encryption methods on the CA Access Control Enterprise Management sever and on the CA Access Control endpoints. All password history will be lost if you select to change the encryption method of existing CA Access Control endpoints.

This command has the following format:

```
ChangeEncryptionMethod.sh [DES|TRIPLEDES|SCRAMBLE|AES128|AES192|AES256]
```

More information:

sechkey Utility—Change the Symmetric Encryption Method (see page 110)

dbmgr Utility

The dbmgr utility lets you create, manage, and maintain the CA Access Control database files.

Note: This utility replaces the following utilities from previous versions: dbdump, rdbdump, dbutil, secredb, sedb2scr, and semigrate.

Important! Use this utility only with the guidance of support personnel during problem resolution. For assistance, contact CA Support at http://ca.com/support.

To run the dbmgr utility, you must have the ADMIN, AUDITOR, or SERVER attribute.

The utility handles several tasks and has the associated following functions:

Task	Function
Create a database (see page 28)	`dbmgr -create`
Display database information (see page 31)	`dbmgr -dump`
Create a script that defines a database (see page 33)	`dbmgr -export`
Copy database data to a flat file (see page 34)	`dbmgr -migrate`
Manage an existing database (see page 36)	`dbmgr -util`
Backup a database (see page 38)	`dbmgr -backup`
Restore a database (see page 39)	`dbmgr -restore`

dbmgr -create Function—Create a Database

The dbmgr -create function generates a new empty database. Use this function only at installation time, or when you want to create a database or PMDB. CA Access Control creates the database in the current directory.

Note: If you want to add user-defined classes to the new database, first run the seclassadm utility after creating the new database.

This command has the following format:

```
dbmgr {-create|-c} {-c[q]|-h} [-d] [-f filename] \
    [-n] [-o] [-t terminalNames] \
    [-u userName [,userName...]] [-ux userName [,userName...]]\
    [-v] [-w] [-k] [-n pathName]
```

-create|-c

Executes the database creation function of the dbmgr utility.

-c

Prompts you for whether you want to create a new database.

-cq

Creates a new database without prompting you first.

-h

Displays the help for this function.

-d

Prints database layout documentation. The output contains a full description of the structure and property formats used in the database.

-f *filename*

Defines a file to direct output to, instead of th standard output device.

-k

Specifies to run the coexistence utility when the database creation completes.

-n *pathName*

(UNIX Only). Defines the full pathname of the CA Access Control database to back up.

When you are creating a new database, a basic class scheme is generated. When you are adding new classes to the database using the seclassadm utility, the class information is stored in a file in the database directory. To back up a specific database with its class scheme (such as a policy model database), specify its location with the -n option. The user-defined class information is taken from that location. If you do not specify the -n option, the class information file is searched for in the local directory were the database is to be created. If it is not found there, the file is taken from the active CA Access Control security database directory.

-o

Adds Unicenter TNG classes to an existing database.

-t *terminalName*

Defines a comma-separated list of terminals, from which the superusers can manage the local database, to create in the database.

-u *userName [,userName...]*

Defines a comma-separated list of users to create in the database. These users are defined as CA Access Control security administrators.

If the -t option is specified, these users are authorized to manage the local database from the defined terminals.

See also the -ux parameter.

-xu *userName [,userName...]*

Defines a comma-separated list of enterprise users to be defined as CA Access Control security administrators.

If the -t option is specified, these users are authorized to manage the local database from the defined terminals.

If no users are created dbmgr -create creates a user in the database that corresponds to *root* on UNIX, or Administrator on Windows, with the ADMIN, AUDITOR, and IGN_HOL attributes.

-v

Disables the progress messages.

-w

Creates a new database that includes Unicenter TNG classes.

Note: The -v and -d options cannot be used together.

Example: Create a new database on Windows

If at the system prompt c:\temp>, enter the following command:

```
dbmgr -c -c -u user1 -t myterminal.company.com
```

When you confirm that you want to create the database, the utility creates a new database in the c:\temp directory. It creates the user *user1* in the database, who has the ADMIN, AUDITOR, and IGN_HOL attributes, and can administer the database from the terminal *myterminal.company.com*.

Example: Create a new database on UNIX

If at the \tmp\db directory, enter the following command:

```
dbmgr -c -cq -d -f dbLayout
```

The utility creates a new database in the \tmp\db directory. It also creates a file (dbLayout) that contains the database layout documentation. By default, it creates the user *root* in the database, and assigns it the ADMIN, AUDITOR, and IGN_HOL attributes.

More information:

seclassadm Utility—Administer CA Access Control Classes (see page 116)
eACoexist Utility—Detect and Register Coexisting Trusted Programs (see page 50)

dbmgr -dump Function—Display Database Information

The dbmgr -dump function reports on the records in the database. Use this function to perform the following operations:

- Display information for records of a specified class
- Display information for a single record of a specified class
- Display information for all records of a class, except a specified one
- Generate lists of classes and property definitions
- Generate a list of groups that a user belongs to
- Generate a list of records of a particular class

This function assumes that the CA Access Control daemons are not running; you must invoke it from the directory where the database resides. If you use the -r switch, CA Access Control daemons must be running, and you must have the ADMIN, AUDITOR, or SERVER attribute. To execute this function, you must also have READ and WRITE permission on the database files.

This command has the following format:

```
dbmgr {-dump|-d} [-h] [-r] [-f fileName] \
    [c] [fc] [g user] [l class] [p class] [fp class] \
    [d class [props|@fileName] \
    [dn class [props|@fileName] \
    [e class record [props|@fileName] \
    [en class record [props|@fileName] \
    [o class record [props|@fileName] \
    [on class record [props|@fileName]
```

-dump|-d

Executes the database dump function of the dbmgr utility.

-f *fileName*

Directs output to the specified file, instead of the standard output device.

-h

Displays the help for this function.

-r

Displays information about the database currently being used by the authorization daemon.

If you omit this option, dbmgr displays information about the database in the current directory.

c

Lists the names of all classes defined in the database.

d *class* [*props*|*@fileName*]

Displays the values of selected properties for all records of a class. The *class* parameter specifies the class. The *props* parameter defines a space-separated list of properties whose values you want to display.

To read the property list from a file, specify the full pathname of a file, preceded by an “at” sign (@). Each property listed in the file must appear on a separate line.

If you do not specify any properties, the values of all the properties are listed.

dn *class* [*props*|*@fileName*]

Same as the *d* option, only properties with unknown values are not displayed.

e *class record* [*props*|*@fileName*]

Displays the values of selected properties for all records of a class *except* for a single, specified record. The *class* parameter specifies the class. The *record* parameter specifies the name of the record to omit from the list. The *props* parameter defines a space-separated list of properties whose values you want to display.

To read the property list from a file, specify the full pathname of a file, preceded by an “at” sign (@). Each property listed in the file must appear on a separate line.

If you do not specify any properties, the values of all the properties are listed.

en *class record* [*props*|*@fileName*]

Same as the *e* option, only properties with unknown values are not displayed.

fc

Lists all class information for all classes in the database.

fp *class*

Lists all property information on properties of the specified class.

g *user*

Lists the groups that the specified user is a member of.

l *class*

Lists all the records in the specified class.

o *class record property* / on *class record property*

Displays the values of selected properties for a single record of a class. The *class* parameter specifies the class. The *record* parameter specifies the record. The *props* parameter defines a space-separated list of properties whose values you want to display.

To read the property list from a file, specify the full pathname of a file, preceded by an “at” sign (@). Each property listed in the file must appear on a separate line.

If you do not specify any properties, the values of all the properties are listed.

o *class record property* / on *class record property*

Same as the *o* option, only properties with unknown values are not displayed.

p *class*

Lists the names of the properties of the specified class.

Note: You can only specify one other option apart from -r and -f.

dbmgr -export Function—Create Script that Defines a Database

The dbmgr -export function replicates a database on other stations. It generates a script that consists of the selang commands required to define an existing database.

Note: You cannot copy database files from one architecture to another when using native commands (such as cp or tar on UNIX or copy on Windows), if the files do not use the same byte order. For example, you cannot copy a database from a Sparc-based machine to an Intel based machine, because each uses a different byte order.

Important! Review the script before you execute it.

This command has the following format:

`dbmgr {-export|-e} {-l|-r} [-c` *`className`*`] [-f` *`fileName`*`]`

-export|-e

Executes the database export function of the dbmgr utility.

-h

Displays the help for this function.

-l

Exports the database in the current directory.

Note: This option assumes the CA Access Control daemons are not running. If the daemons are running, then it assumes you are operating on a different database from the one being used by the daemons.

-r

Exports the database currently being used by CA Access Control. You must have the ADMIN or SERVER attribute, and the CA Access Control daemons must be running.

-c *className*

Defines a space-separated list of classes which you want to export from the database.

-f *fileName*

Directs output to the specified file, instead of the standard output device. You can then create a new database from the file, by instructing selang to read the commands from the file.

dbmgr -migrate Function—Copy Data to a Flat File

The dbmgr -migrate function copies data from user and program records in an existing database to a flat file (binary format). It can also copy the data from the flat file into a new database. The database from which the data is imported must be version 1.21 or later.

When you copy a flat file into a new database, use the same version of this function that you used to create the flat file. If you have more than one version, we strongly recommend that you use the most recent version.

Note: For better security, delete the old database, the script used to build the new database, and the flat file created by this function after copying the data from the old database into the new database.

Important! Always create a backup of the database before using this function.

This command has the following format:

```
dbmgr {migrate|-m} {-r|-w|-h} [-s] filename \
    [-v versionNumber] [-f fileName]
```

-migrate|-m

Executes the database migration function of the dbmgr utility.

filename

Defines the flat file you want to copy data from or into.

-f *filename*

Directs output to the specified file, instead of the standard output device.

-h

Displays the help for this function.

-r

Reads the database in the current directory and copies certain data into the flat file *filename*.

-s

Reads the information from the database using the CA Access Control server rather than reading the database directly. This option is only valid with the -r switch.

You must have administrator privileges and R (read) and W (write) access to the terminal to use this option.

If you do not specify this option, the function reads from or writes to the database in the current directory.

-v *versionNumber*

Reads a flat file that was created by a previous version. This option is only valid for -w command. Enter this option after the file name and supply the version number.

-w

Reads the flat file *filename* and copies the data into the database in the current directory.

Example: Copy data from an existing database to a new database

The following steps illustrate how to copy data from an existing database into a new database. The old database is assumed to be in the directory /tmp/old_db. The new database is assumed to be in the directory *ACInstallDir*/seosdb (where *ACInstallDir* is the directory in which you installed CA Access Control).

Note: This procedure is written using UNIX pathnames but can be followed on Windows by modifying these pathnames as appropriate.

1. Log in as a superuser.
2. If the CA Access Control daemons are running, shut them down with the following command:

   ```
   secons -s
   ```

3. Create a backup of the old database by copying it to a different location or to a backup medium.
4. Copy the database into /tmp/old_db, then create a script that duplicates the old database by running the dbmgr utility on the old database:

   ```
   cd /tmp/old_db
   /opt/CA/AccessControl//bin/dbmgr -export -l -f lang_script
   ```

5. Create a new database:

   ```
   cd /opt/CA/AccessControl//seosdb
   /opt/CA/AccessControl//bin/dbmgr -c -cq
   ```

6. Execute the script generated in the previous step and create the new database:

   ```
   cd /opt/CA/AccessControl//seosdb
   /opt/CA/AccessControl//bin/selang -l /tmp/old_db/lang_script
   ```

7. Execute the dbmgr utility to create a flat file containing data from the old database:

   ```
   cd /tmp/old_db
   /opt/CA/AccessControl//bin/dbmgr -migrate -r flat_file
   ```

8. Load the data from the flat file into the new database:

   ```
   cd /opt/CA/AccessControl//seosdb
   /opt/CA/AccessControl//bin/dbmgr -migrate -w /tmp/old_db/flat_file
   ```

dbmgr -util Function—Manage Existing Database

The dbmgr -util function performs management and maintenance operations on a database. It assumes CA Access Control is not currently running. Invoke it from the directory where the database resides.

The -util option is used to manage and manipulate the local database specified by the parameter *filename*. Database files have the extension .dat and must be DBIO files. Database index files (files with the extension .001) cannot be used with the -util option.

This command has the following format:

```
dbmgr {-util|-u} [-h] \
    [-all filename] \
    [-build filename] \
    [-check] \
    [-close] \
    [-dump filename] \
    [-dup src dst] \
    [-fast] \
    [-free filename] \
    [-index filename] \
    [-key filename] \
    [-load db ascii \
    [-scan filename] \
    [-scana filename] \
    [-stat filename] \
    [-verify] \
    [-f fileName]
```

-util-u

Executes the database management and maintenance functions of the dbmgr utility.

-all *filename*

Performs all index checks; same as specifying the *-index* and *-free* options.

-build *filename*

Builds indexes of a DBIO based on data records.

-check

(UNIX only). Performs a fast sanity and consistency check on all index entries for all database files.

-close

Closes the database if it is open.

-dump *filename*

Dumps the data file as ASCII on the standard output device.

-dup *src dst*

Duplicates the DBIO file based on the file header.

-f *fileName*

Directs output to the specified file, instead of the standard output device.

-fast

Performs a fast sanity check on all index entries for all the database files.

-free *filename*

Checks for a free index.

-index *filename*

Checks the consistency of the index.

-key *filename*

Sequentially scans an index file.

-load *db* ascii

Loads an ASCII file and converts it into a DBIO file.

-scan *filename*

Scans the database sequentially.

-scana *filename*

Scans the database sequentially, including deleted records.

-stat *filename*

Lists the header information of the database file.

-verify

(UNIX only). Verifies that certain predefined objects exist in the database; for example, SEOS, ADMIN, and UACC for all classes.

dbmgr -backup Function—Back Up a Database

The dbmgr -backup function creates an online backup of the CA Access Control database in the specified directory. This function is available whether the CA Access Control daemons are running or not.

This command has the following format:

`dbmgr {-backup|-b} backup_directory`

-backup|-b

Executes the database backup function of the dbmgr utility.

backup_directory

Defines the backup directory. This directory cannot be located on a remote machine; if the directory does not exist, the function creates it.

dbmgr -restore Function—Restore a Database

Valid on UNIX

The dbmgr -restore function performs an online restore of the CA Access Control database in the specified directory. This function is available whether the CA Access Control daemons are running or not.

This command has the following format:

`dbmgr {-restore|-r} restore_directory`

-restore|-r

Executes the database restore function of the dbmgr utility.

restore_directory

Defines the directory where the database you want to restore resides.

defclass Utility—Define User-defined Asset Types as Classes

CA Access Control defines basic Unicenter TNG asset types in each CA Access Control database and every new PMDB that is defined. The defclass script defines user-defined security asset types as CA Access Control classes in the CA Access Control database.

Note: The installation program automatically executes this script when Unicenter Integration is selected. It can, and should, be called manually whenever a new PMDB is created.

This command has the following format:

`defclass`

Note: On UNIX, this utility is supplied as a script file; you need to specify the .sh extension to run it. It is only available if you enable Unicenter integration, which is disabled by default.

DictImport Utility—Import the Dictionary File

The DictImport utility prepares and imports dictionary files into the CA Access Control database. After installing CA Access Control, you must import the dictionary file into the CA Access Control database and then activate it, so you can set password protection.

The DictImport utility sets the use_dbdict password rule to *db* and activates the DICTIONARY class and PASSWORD class.

Note: The centralized dictionary is disabled if the PASSWORD class is not active.

This command has the following format:

```
DictImport [-h] [-o selangFilename] [-f dictionaryFilename]
```

Note: This utility is supplied as a script file and is located in the lbin directory.

-f *dictionaryFilename*

Generates selang commands that import all the dictionary words from the specified file. If you omit this option, the dictionary file is defined from values in the configuration settings.

-h

Displays the help for this utility.

-o *selangFilename*

Writes selang commands to the specified file. If you omit this option, selang commands are written to the standard output device.

dmsmgr Utility

The dmsmgr utility lets you manage the advanced policy management infrastructure. Infrastructure components include CA Access Control endpoints, Deployment Map Server (DMS) and Distribution Host (DH).

The utility handles several tasks and has the associated following functions:

Task	Function
Create a DMS or a DH (see page 41)	`dmsmgr -create`
Remove a DMS or a DH (see page 42)	`dmsmgr -remove`

Task	Function
Remove obsolete nodes from the DMS database (see page 43)	`dmsmgr -cleanup`
Configure advanced policy management (see page 44)	`dmsmgr -config`
Restore a DMS or a DH (see page 45)	`dmsmgr -restore`
Synchronize a DMS or a HD (see page 46)	`dmsmgr -sync`

dmsmgr -create Function—Create a DMS or a DH

The dmsmgr -create function creates a Deployment Map Server (DMS) or a Distribution Host (DH) on a computer where CA Access Control is installed.

Note: You can also create a DMS or a DH during installation.

Note: The user running the utility is always granted administration rights for the created DMS or DH.

This command has the following format:

```
dmsmgr -create -auto [-osgroups] [-admin user [,user...]] [-xadmin user [,user...]]
\
[-desktop hosts]

dmsmgr -create -dms name \
[-admin user [,user...]] [-xadmin user [,user...]] \
[-desktop hosts] [-subscriber dh-names]

dmsmgr -create -dh name [-parent dms_name@hostname] \
[-admin user [,user...]] [-xadmin user [,user...]] \
[-desktop hosts]
```

-admin *user* [,*user*...]]

(Optional) Defines internal users as administrators of the created DMS or DH.

-auto

Creates a DMS and a DH with default names (DMS__, DH__, and DH__WRITER).

Use this option to easily create a DMS and a DH and the required associations between them.

-osgroups

(Optional) Specifies to create predefined host groups when you create a DMS.

-desktop *hosts*

(Optional) Defines a comma-separated list of computers that have TERMINAL access rights to the computer with the created DMS or DH.

Note: Whether specified or not, the terminal running the utility is always granted administration rights for the created DMS or DH.

-dh *name*

Creates a DH with the *name* specified on the local host.

Note: If you use this option to create a DH, CA Access Control lets you know that you then need to synchronize the DMS and DH even if the DH is already subscribed and no policies were previously sent. This message is a reminder of the steps you need to take and may not be indicative of the actual situation. If you completed all required steps, you can safely ignore it.

-dms *name*

Creates a DMS with the *name* specified on the local host.

-parent *dms_name@hostname*

(Optional) Defines a DMS that the created DH will send endpoint notifications to. Specify the DMS in the following format: *DMS_name@hostname*.

-subscriber *dh_names*

(Optional) Defines a comma-separated list of DH PMDBs that the created DMS will policy updates to. Specify each DH in the following format: *DH_name@hostname*.

-xadmin *user* [,*user*...]]

(Optional) Defines enterprise users as administrators of the created DMS or DH.

dmsmgr -remove Function—Remove a DMS or a DH

The dmsmgr -remove function removes a DMS or a DH on a computer where CA Access Control is installed.

This command has the following format:

```
dmsmgr -remove {-dms|dh} name
```

```
dmsmgr -remove -auto
```

-auto

Removes the default DMS and DH from the local host.

These are the DMS and DH databases created by default during installation or when you use dmsmgr -create -auto.

-dh *name*

Removes the specified *name* DH from the local host.

-dms *name*

Removes the specified *name* DMS from the local host.

dmsmgr -cleanup Function—Remove Obsolete Nodes

The dmsmgr -cleanup function removes obsolete nodes from the DMS or DH database. These are HNODE objects that represent CA Access Control nodes that have been unavailable for a specified amount of time.

Note: As a routine maintenance procedure, you should clean the DMS and DH from these obsolete nodes.

This command has the following formats:

```
dmsmgr -cleanup {-hnode|-deployment} -days number {-dms|-dh} name
```

```
dmsmgr -cleanup -policy name -vcount number {-dms|dh} name
```

-hnode

Removes HNODE objects that represent CA Access Control nodes that have been unavailable for more than *number* days.

-deployment

Removes the DEPLOYMENT objects that are older than *number* days.

-policy *name*

Removes the POLICY objects (policy versions) that belong to the specified policy and are older than *number* versions.

-dh *name*

Defines the name of the DH you want to remove the obsolete nodes from.

-dms *name*

Defines the name of the DMS you want to remove the obsolete nodes from.

-vcount

Defines the number of versions to keep.

dmsmgr -config Function—Configure Advanced Policy Management

The dmsmgr -config function configures advanced policy management.

This command has the following format:

```
dmsmgr -config[-] [host_name] {-endpoint|-dhname names|-drname names}

dmsmgr -config -osgroups [-dms name]
```

-config[-]

Configures or removes the configuration of advanced policy management.

-dhname *names*

Configures the endpoint to work with the comma-separated list of Distribution Hosts.

-dms *name*

Defines the name of the DMS on which the automatic host groups are created.

-drname *names*

Configures the endpoint to work with the comma-separated list of disaster recovery Distribution Hosts.

-endpoint

Configures the endpoint for advanced policy management.

host_name

Performs the configuration on *host_name*. If no host is specified, performs the configuration on the local computer.

-osgroups

Adds automatic host groups to the DMS.

Note: For more information about automatic host groups, see the *Enterprise Administration Guide*.

dmsmgr -restore Function—Restore a DMS or DH

The dmsmgr -restore function restores a DMS or a DH from backup files. You can restore a DMS or DH when CA Access Control is running or stopped, over an existing DMS, or into a new directory.

This command has the following format:

```
dmsmgr -restore -dms name -source path\
[-replica name|-parent name] [-subscriber dhname[,dhname...]]\
[-admin user[,user...]] [-xadmin user[,user...]]

dmsmgr -restore -dh name -source path\
[-parent name] [-admin user[,user...]]\
[-xadmin user[,user...]] [-desktop host[,host...]]
```

-admin *user*[,*user*...]

(UNIX) Defines internal users as administrators of the restored DMS or DH.

-desktop *host*[, *host*...]

(Optional) Defines a list of computers that have TERMINAL access rights to the computer with the restored DH.

Note: Whether specified or not, the terminal running the utility is always granted administration rights for the restored DH.

-dh *name*

Defines the name of the DH that is restored on the local host.

-dms *name*

Defines the name of the DMS that is restored on the local host.

-parent *name*

(Optional) Defines the name of the subscriber's parent. Use this parameter if you have set up CA Access Control in a disaster recovery deployment and you restore a disaster recovery DMS or a DH. If you restore a disaster recovery DMS, specify the name of the production DMS; if you restore a DH, specify the name of the parent DMS. Specify the parent in the following format: *name@hostname*.

-replica *name*

(Optional) Defines the name of the disaster recovery DMS. Use this parameter if you have set up CA Access Control in a disaster recovery deployment and restore a production DMS. Specify the disaster recovery DMS name in the following format: *DMS_name@hostname*.

-source *path*

Defines the directory that contains the backup files to restore.

-subscriber *dh_name*[, *dh_name...*]

(Optional) Defines a comma-separated list of DHs that the restored DMS will send policy updates to. Specify each DH in the following format: *DH_name@hostname*.

-xadmin *user*[,*user...*]

(UNIX) Defines enterprise users as administrators of the restored DMS or DH.

dmsmgr -sync Function—Synchronize a DMS or a DH

The dmsmgr -sync function synchronizes between the DMS and the DH to create a mirror image of the DMS on the DH. You can execute the synchronization process from the Enterprise Management Server or from a dedicated Distribution Server.

This command has the following format:

dmsmgr -sync -dhname <*dhname*> [-dms<*dms-name*>]

dmsmgr -sync self [-dh<*dhname*>]

-dms<*dms-name*>

Synchronizes the DMS with the DH.

Note: Run the command from the DMS computer.

-dh<*dhname*>

Synchronizes the DH with the DMS.

Note: Run the command from the DH computer.

-dhname<*name*>

Specifies a comma separated list of Distribution Hosts.

self

Specifies to synchronize the DH with the DMS.

Note: Run the command from the Distribution Server.

eacpg_gen Utility—Define Best Practice Policies

Valid on Linux

eacpg_gen is also known as Policy Generator. This menu-driven utility provides an easy method to define a policy for CA Access Control applications. Policy Generator can be used on a test system that has no CA Access Control rules in it. It aims to protect enterprise applications and/or operating systems and their confidential data by applying security best practices around those critical electronic assets.

Application cells are created with a “default-deny” paradigm. These policies are similar to the concept of a UNIX chroot() jail. When such a policy is generated for an Internet-facing application, the risk of host compromise using that application is greatly reduced.

An application cell is an access control list (ACL) rule that blocks an application. For each application, eacpg_gen generates a number of application cells. The application cell enforces access to specific resources only. Any process protected with a cell policy cannot access resources it has not specifically been given access to in the policy. This keeps would-be attackers from writing to unauthorized areas of disk or executing unauthorized binaries.

Note: Verify that the secadmin and group secadmins exists in the database before you run this utility.

Policy generation has several key steps:

- Initialization
- Application inspection
- Application testing
- Policy generation
- Applying the policy
- Testing the policy

This command has the following format:

```
eacpg_gen \
    [-u user] \
    [-g group] \
    [-p path] \
    [-o owner] \
    [-w wheel] \
    [-m machine] \
    [-a] \
    [-s file] \
    [-# step] \
    [-x]
```

-u *user*

Specifies the user for the process to run as.

-g *group*

Specifies the group name that will own the process.

-p *path*

Specifies the full path to the program.

-o *owner*

Specifies the policy owner.

-w *wheel*

Sets as 'secadmins' group (recommended).

-m *machine*

Specifies the machine name.

-a

Sets whether to apply the generated rules.

-s *file*

Specifies the full path and the file name to save the policy rules.

-# *step 1-2*

Should be set to 2.

-x

Toggles between warn and fail mode.

Example: Run the Policy Generator

1. (Initialization). Execute the policy generator:

   ```
   eacpg_gen
   ```

2. Type **y** at the prompt to place the system into Warning Mode.
3. Supply the policy generator with the full path to the executable, for example:

   ```
   /work/WebServers/apache_1.3.26/bin/htppd
   ```

4. Accept the default user name.
5. Accept the default group name.
6. Type **y** at the prompt to verify that the information is correct.

   (Application inspection). The policy generator begins to collect data on the process you are creating a policy for.

7. Verify the information on the screen and press Enter.
8. (Application testing). Start the application. For example:

   ```
   ./apachectl start
   ```

9. Stop the application. For example:

   ```
   ./apachectl stop
   ```

   Note: At this point after you have started and stopped the application. It is best to start it again and allow for normal usage data to be collected. You can allow this inspection to take place for as long as you would like; the longer it runs the more data the policy generator can collect and the more accurate the resulting policy will be. When you feel you have collected enough data, continue to the next step.

10. (Policy generation). Save the policy to a file (enter *filename*.txt and press Enter).
11. (Policy application). Type **y** to apply the policy.

12. Type **y** to put the system into *Fail* mode and begin policy enforcement.
13. (Policy testing). Test the policy.

    Following is a sample screen showing a policy test on a file named evil.html.

```
Linux:/srv/www/htdocs: #telnet localhost 80
Trying ::1...
telnet: connect to address ::1: Connection refused
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
GET /evil.html
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML><HEAD>
<TITLE>403 Forbidden</TITLE>
<HEAD><BODY>
<H1>Forbidden</H1>
You don't have permission to access /evil.html
on the server. <P>
<HR>
<ADDRESS>Apache/1.3.26 Server at linux.local Port 80</ADDRESS>
</BODY></HTML>
Connection closed by foreign host.
Linux:/srv/www/htdocs# []
```

    Now that the policy is applied, the file evil.html is no longer available. This is because it was outside the scope of our normal usage profile.

eACoexist Utility—Detect and Register Coexisting Trusted Programs

Valid on Windows

The eACoexist utility detects any coexisting programs in the local system (for example, CA Anti-Virus). If the detected program is trusted, CA Access Control registers the program using a SPECIALPGM rule. A special program rule defines the types of access for that program and makes sure that CA Access Control bypasses it when granting access.

This command has the following format:

`eACoexist [`*`plug-in-path`*`]`

plug-in-path

(Optional) Defines the path to the folder that contains the coexistence plug-ins you want the coexistence program to use.

If you do not define a path, the program uses the default path where the coexistence plug-ins are stored (*ACInstallDir*/Coexistence).

More information:

How the Coexistence Utility Works (see page 51)
response.ini—Configure the Coexistence Utility (see page 65)

How the Coexistence Utility Works

The coexistence utility (eACoexist) that CA Access Control supplies, lets you resolve potential conflicts with other programs on the local computer. To understand what CA Access Control does to resolve these potential conflicts, and to be able to affect how these conflicts are resolved, you need to understand how the utility works.

When the coexistence utility runs, it performs the following actions:

1. Checks that *one* of the following conditions apply:
   a. CA Access Control is not running.
   b. You have the ADMIN attribute.

   If neither conditions apply, the utility exits.
2. Locates the response.ini file, as follows:
   - When the utility runs during installation, it uses the path *media_drive*:\Coexistence_*architecture*
   - If CA Access Control is installed on the computer, it uses the following registry key value:

     `HKLM\SOFTWARE\ComputerAssociates\AccessControl\AccessControl\SeOSD\ResponseFile`

   If the file does not exist the utility exits.
3. Locates the coexistence plug-ins directory, as follows:
   - If you run the utility and pass a parameter from the command line, it uses this as the plug-ins' path.
   - When the utility runs during installation, it uses the path *media_drive*:\Coexistence_*architecture*
   - If you run the utility with no parameters, it concatenates the string "\Coexistence" to the following registry key value:

     `HKLM\SOFTWARE\ComputerAssociates\AccessControl\AccessControl\SeOSPath`

   If the directory does not exist, or there are no coexistence plug-ins in the directory, the utility exits.

4. Executes the discovery process.

   To do this, it enumerates the executables in the coexistence plug-ins directory and executes them one by one, as follows:

   a. Stores the result of the plug-in execution in %windir%\EACDiscovery.ini

      Note: The utility automatically deletes this file on successful completion of the plug-in discovery process.

   b. Checks that the output file EACDiscovery.ini exists.

      If the file does not exist, the utility continues to execute the next plug-in.

   c. For each product section in EACDiscovery.ini, concatenates the section (product) name and version value and checks whether the response file contains the matching section.

      Note: The response.ini file contains a section for each coexisting program. If a section name appears with a version number, for example, eTrust Audit-1.5, the utility performs the action only for the specified version.

   d. If a matching section exists in the response file, executes the action that is set by the value of the Act-Utility-0 in that section, as follows:

      - **1**—Issues a warning that the discovered product is not compatible with CA Access Control.
      - **2**—Stops the discovered product's services.

        The utility retrieves the discovered product's services from the EACDiscovery.ini file.
      - **3**—Same as 2, but during CA Access Control installation.
      - **4**—Starts the discovered product's services.

        The utility retrieves the discovered product's services from the EACDiscovery.ini file.
      - **5**—Creates trusted program rules (SPECIALPGM) for the discovered product's processes and starts CA Access Control.

        The utility retrieves the discovered product's processes from the EACDiscovery.ini file. It also retrieves the respective program type (pgmtype) from this file. It then creates a temporary script file (*ACInstallDir*\Data\discoveryscp) that it executes when CA Access Control starts.
      - **6**—Same as 2, but during CA Access Control uninstall.

      Note: Each section can contain more than one action. For example, you can have Act-Utility-0, Act-Utility-1, and Act-Utility-2 that are executed in that order.

More information:

How the Policy Manager Plug-In Works

The coexistence utility runs the Policy Manager plug-in to scan the computer for Policy Manager registry keys and executables before the CA Access Control installation begins, as follows:

- Queries the following registry key for existence:

  `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths\SeAM.Exe`

  If the registry key exists, the plug-in:

  - Reads the value of the Path entry
  - Returns the following executable pathname:

    `FilePathFromRegistry``\Bin\SeAM.exe`

  - Issues a compatibility warning during CA Access Control installation

    This is the default action as defined in the response file. The Policy Manager plug-in does not add a trusted program (SPECIALPGM) rule by default.

Note: The response file determines the default action that the coexistence utility performs at prescribed stages (what to do before CA Access Control installation, after CA Access Control installation, and so on). Also, the Policy Manager application is no longer provided with CA Access Control.

How the BrightStor Plug-In Works

The coexistence utility runs the BrightStor plug-in to scan the computer for CA BrightStor registry keys and executables at the end of a CA Access Control installation and whenever the utility runs, as follows:

1. Queries the following registry keys for existence:

   ```
   HKLM\SOFTWARE\ComputerAssociates\BrightStor ARCserve
   Backup\UniversalClientAgent\Common
   HKLM\SOFTWARE\ComputerAssociates\Cheetah\UniversalClientAgent\Common
   HKLM\SOFTWARE\ComputerAssociates\BrightStor Enterprise
   Backup\UniversalClientAgent\Common
   ```

   When the first registry key exists, the plug-in:

   - Reads the value of the Path entry
   - Returns the following executable pathname:

     FilePathFromRegistry\UnivAgent.exe

   - Creates a SPECIALPGM resource of type DCM

     This is the default action as defined in the response file.

2. If the plug-in cannot find any of the registry keys in Step 1, it queries the following registry keys for existence:

   ```
   HKLM\SOFTWARE\ComputerAssociates\BrightStor ARCserve Backup\Base\Path
   HKLM\SOFTWARE\ComputerAssociates\Cheetah\Base\Path
   HKLM\SOFTWARE\ComputerAssociates\BrightStor Enterprise Backup\Base\Path
   ```

   When the first registry key exists, the plug-in:

   - Reads the value of the HOME entry
   - Returns the following executable pathname:

     FilePathFromRegistry\carunjob.exe

   - Creates a SPECIALPGM resource of type DCM

     This is the default action as defined in the response file.

3. If the plug-in also cannot find any of the registry keys in Step 2, it queries the following registry key for existence:

   ```
   HKLM\SOFTWARE\ComputerAssociates\ARCserveIT\Base\Path
   ```

   If the registry key exists, the plug-in:

   - Reads the value of the HOME entry
   - Returns the following executable pathname:

     FilePathFromRegistry\ASRunJob.exe

   - Creates a SPECIALPGM resource of type DCM

     This is the default action as defined in the response file.

4. Queries the following registry keys for existence:

   ```
   HKLM\SOFTWARE\ComputerAssociates\CA BAOF\CurrentVersion
   HKLM\SOFTWARE\ComputerAssociates\BrightStor Backup Agent for Open
   Files\CurrentVersion
   ```

   When the first registry key exists, the plug-in:

   - Reads the value of the ServicePath entry
   - Creates a SPECIALPGM resource of type DCM for *ServicePathFromRegistry*

     This is the default action as defined in the response file.

Note: The response file determines the default action that the coexistence utility performs at prescribed stages (what to do before CA Access Control installation, after CA Access Control installation, and so on).

How the Dr. Watson Plug-In Works

The coexistence utility runs the Dr. Watson plug-in to scan the computer for Dr. Watson executables at the end of a CA Access Control installation and whenever the utility runs, as follows:

- Queries the following pathname for existence:

  ```
  %windir%\system32\drwtsn32.exe
  ```

  If the file exists, the plug-in creates a SPECIALPGM resource of type DCM.

  This is the default action as defined in the response file.

Note: The response file determines the default action that the coexistence utility performs at prescribed stages (what to do before CA Access Control installation, after CA Access Control installation, and so on).

How the eTrust AV Plug-In Works

The coexistence utility runs the eTrust AV plug-in to scan the computer for CA Antivirus registry keys and executables at the end of a CA Access Control installation and whenever the utility runs, as follows:

1. Reads the following registry key entry values:

   `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths\InocIT.Exe\Path`
   `HKLM\SOFTWARE\ComputerAssociates\eTrustITM\CurrentVersion\Path\Home`

   If one of the entries returns a value, the plug-in creates the following SPECIALPGM resources of type DCM:

   - *FilePathFromRegistry*\InoRT.exe
   - *FilePathFromRegistry*\InoTask.exe
   - *FilePathFromRegistry*\InocIT.exe
   - *FilePathFromRegistry*\ShellScn.exe

   This is the default action as defined in the response file.

2. Reads the following registry key entry value:

   `HKLM\SOFTWARE\ComputerAssociates\ScanEngine\Path\Engine`

   If the entry returns a value, the plug-in creates the following SPECIALPGM resource of type DCM:

   FilePathFromRegistry\InoCmd32.exe

Note: The response file determines the default action that the coexistence utility performs at prescribed stages (what to do before CA Access Control installation, after CA Access Control installation, and so on).

How the Scout Plug-In Works

The coexistence utility runs the Scout plug-in to scan the computer for SurfControl Web Filter for Windows registry keys and executables at the end of a CA Access Control installation and whenever the utility runs, as follows:

- Queries the following registry key for existence:

  `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths\Scscout.Exe`

  If the registry key exists, the plug-in:

  - Reads the value of the Path entry
  - Returns the following executable pathname:

    FilePathFromRegistry`\scoutsvc.exe`
  - Creates a SPECIALPGM resource of type DCM

    This is the default action as defined in the response file.

Note: The response file determines the default action that the coexistence utility performs at prescribed stages (what to do before CA Access Control installation, after CA Access Control installation, and so on).

How the Unicenter Plug-In Works

The coexistence utility runs the Unicenter plug-in to scan the computer for CA Unicenter registry keys and executables at the end of a CA Access Control installation and whenever the utility runs, as follows:

1. Uses CAUENV.dll to retrieve the path of the CA Unicenter directory (*UniPath*)
2. Creates the following SPECIALPGM resources of type DCM:
   - *UniPath*\Bin\sfauditd.exe
   - *UniPath*\Bin\secdos2.exe
   - *UniPath*\Bin\caulgnd.exe
   - *UniPath*\Bin\sccommit.exe
   - *UniPath*\Bin\dsbulist.exe
   - *UniPath*\Bin\fmpost.exe
   - *UniPath*\Bin\catlbl.exe
   - *UniPath*\Bin\caanal.exe
   - *UniPath*\Bin\cascan.exe
   - *UniPath*\Bin\causamd.exe
   - *UniPath*\Bin\acbrows.exe
   - *UniPath*\Bin\secadmin.exe

- *UniPath*\Bin\dsbufcrt.exe
- *UniPath*\Bin\cnvpwd.exe
- *UniPath*\Bin\fmeng.exe
- *UniPath*\Bin\fmmscan.exe
- *UniPath*\Bin\cadevscn.exe
- *UniPath*\AGENTS\Bin\prfagent.exe
- *UniPath*\AGENTS\Bin\msexchagnt.exe

Note: The response file determines the default action that the coexistence utility performs at prescribed stages (what to do before CA Access Control installation, after CA Access Control installation, and so on).

How the Asset Management Plug-In Works

The coexistence utility runs the Asset Management plug-in to scan the computer for Unicenter DSM services at the end of a CA Access Control installation and whenever the utility runs, as follows:

1. Extracts the directory path of the executable of the service "CA Unicenter NSM Systems Performance Agent for UAM" (*ServicePath*)
2. Creates the following SPECIALPGM resource of type REGISTRY:

   ServicePath\agents\bin\hpacbcol.exe
3. Extracts the directory path of the executable of the service "caf" (*ServicePath*)
4. Creates the following SPECIALPGM resource of type REGISTRY:
   ServicePath\PMAgent\agents\bin\hpacbcol.exe

The coexistence utility also runs the Asset Management plug-in to scan the computer for Unicenter Asset Management version 4 services at the end of a CA Access Control installation and whenever the utility runs, as follows:

1. Extracts the directory path of the executable of the service "CA Unicenter NSM Systems Performance Agent for UAM" (*ServicePath*)
2. Creates the following SPECIALPGM resource of type REGISTRY:

   ServicePath\agents\bin\hpacbcol.exe

The coexistence utility also runs the Asset Management plug-in to scan the computer for Unicenter DSM r11 services at the end of a CA Access Control installation and whenever the utility runs, as follows:

1. Extracts the directory path of the executable of the service "caf" (*ServicePath*)
2. Creates the following SPECIALPGM resource of type REGISTRY:

   ServicePath\PMAgent\agents\bin\hpacbcol.exe

Note: The response file determines the default action that the coexistence utility performs at prescribed stages (what to do before CA Access Control installation, after CA Access Control installation, and so on).

How the Windows Plug-In Works

The coexistence utility runs the Windows plug-in to scan the computer for Windows services and registry keys at the end of a CA Access Control installation and whenever the utility runs, as follows:

1. Extracts the directory path of the executable of the service "WinMgmt" (*ServicePath*)
2. Creates the following SPECIALPGM resource of type REGISTRY:

   ServicePath

   This is the default action as defined in the response file.
3. Creates the following SPECIALPGM resource of type PBF:

   %windir%\System32\cidaemon.exe

   This is the default action as defined in the response file.

Note: The response file determines the default action that the coexistence utility performs at prescribed stages (what to do before CA Access Control installation, after CA Access Control installation, and so on).

How the eTrust Audit Plug-In Works

The coexistence utility runs the eTrust Audit plug-in to scan the computer for eTrust Audit Version 1.5 registry keys and files before the CA Access Control installation begins, as follows:

1. Queries the following registry key for existence:

   `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths\acdistagn.exe`

   `If successful it extracts the "Path" value designated %PathFromRegistry%.`

   If the registry key exists, the plug-in:

   - Reads the value of the Path entry
   - Returns the following executable pathnames:

     FilePathFromRegistry\bin\acactmgr.exe
     FilePathFromRegistry\bin\SeLogRcd.exe
     FilePathFromRegistry\bin\acdistagn.exe
     FilePathFromRegistry\acdistsrv.exe
     FilePathFromRegistry\acfwrecd.exe
     FilePathFromRegistry\acrecorderd.exe
     FilePathFromRegistry\aclogrd.exe
     FilePathFromRegistry\portmap.exe
     FilePathFromRegistry\SeLogRec.exe
     FilePathFromRegistry\SeLogRd.exe
     FilePathFromRegistry\snmprec.exe

     This is the default action as defined in the response file. The eTrust Audit plug-in does not add a trusted program (SPECIALPGM) rule by default.

2. Stops the following services:

   - "eAudit Action Manager"
   - "eAudit Distribution Agent"
   - "eAudit Log Router"
   - "eAudit Recorder"
   - "eAudit Redirector"
   - "eAudit Portmap"

   If a more recent version of eTrust Audit is installed, it stops the following services:

   - "eTrust Audit Action Manager"
   - "eTrust Audit Collector"
   - "eTrust Audit Distribution Agent"
   - "eTrust Audit Distribution Server"
   - "eTrust Audit FW-1 Recorder"

- "eTrust Audit Generic Recorder"
- "eTrust Audit Log Router"
- "eTrust Audit Portmap"
- "eTrust Audit Recorder"
- "eTrust Audit Redirector"
- "eTrust Audit SNMP Recorder"

3. When the CA Access Control installation completes, it restarts these same services.

The coexistence utility also runs the eTrust Audit plug-in to:

- Stop the eTrust Audit services when you uninstall CA Access Control
- Start the eTrust Audit services after the CA Access Control uninstall completes

Note: The response file determines the default action that the coexistence utility performs at prescribed stages (what to do before CA Access Control installation, after CA Access Control installation, and so on).

How the eTrust Audit80 Plug-In Works

The coexistence utility runs the eTrust Audit80 plug-in to scan the computer for eTrust Audit r8 registry keys and files at the end of a CA Access Control installation and whenever the utility runs, as follows:

- Queries the following registry key for existence:

  `HKLM\SOFTWARE\ComputerAssociates\eTrust Audit\Paths`

  If successful it extracts the "Path" value designated %PathFromRegistry%.

  If the registry key exists, the plug-in:

  - Reads the value of the RootPath entry
  - Creates the following SPECIALPGM resources of type DCM:

    FilePathFromRegistry\bin\acactmgr.exe
    FilePathFromRegistry\bin\SeLogRcd.exe
    FilePathFromRegistry\bin\acdistagn.exe
    FilePathFromRegistry\acdistsrv.exe
    FilePathFromRegistry\acfwrecd.exe
    FilePathFromRegistry\acrecorderd.exe
    FilePathFromRegistry\aclogrd.exe
    FilePathFromRegistry\portmap.exe
    FilePathFromRegistry\SeLogRec.exe

FilePathFromRegistry\SeLogRd.exe
FilePathFromRegistry\snmprec.exe

This is the default action as defined in the response file.

Note: The response file determines the default action that the coexistence utility performs at prescribed stages (what to do before CA Access Control installation, after CA Access Control installation, and so on).

How the F-Secure Antivirus Plug-In Works

The coexistence utility runs the F-Secure Antivirus plug-in to scan the computer for F-Secure Anti-Virus registry keys and files, as follows:

- Before the CA Access Control installation begins the plug-in stops the F-Secure Anti-Virus services.
- When the CA Access Control installation completes, the plug-in queries the following registry key for existence:

  HKLM\SOFTWARE\Data Fellows\F-Secure\Anti-Virus

  If successful it extracts the "Path" value designated %PathFromRegistry%.

  If the registry key exists, the plug-in:

  - Reads the value of the Path entry
  - Creates the following SPECIALPGM resources of type DCM:

    FilePathFromRegistry\fssm32.exe
    FilePathFromRegistry\fsgk32st.exe

    This is the default action as defined in the response file. The eTrust Audit plug-in does not add a trusted program (SPECIALPGM) rule by default.
- Whenever the coexistence utility runs, the plug-in:

  a. Stops the F-Secure Anti-Virus services

  b. Creates the same SPECIALPGM resources it creates when the CA Access Control installation completes (as described earlier in this topic)

  c. Restarts the F-Secure Anti-Virus services

Note: The response file determines the default action that the coexistence utility performs at prescribed stages (what to do before CA Access Control installation, after CA Access Control installation, and so on).

How the McAfee VirusScan Plug-In Works

The coexistence utility runs the McAfee VirusScan plug-in to scan the computer for the McAfee VirusScan service at the end of a CA Access Control installation and whenever the utility runs, as follows:

1. Extracts the directory path of the executable of the service "McShield" (*ServicePath*)
2. Creates the following SPECIALPGM resource of type DCM:

   ServicePath

   This is the default action as defined in the response file.

Note: The response file determines the default action that the coexistence utility performs at prescribed stages (what to do before CA Access Control installation, after CA Access Control installation, and so on).

How the Windows Modules Installer Plug-In Works

The coexistence utility runs the Windows Modules Installer plug-in to scan the computer for the Windows Modules Install service at the end of a CA Access Control installation and whenever the utility runs, as follows:

1. Extracts the directory path of the executable of the service "TrusterInstaller" (*ServicePath*)
2. Creates the following SPECIALPGM resource of type PBF:

   ServicePath

   This is the default action as defined in the response file.

Note: The response file determines the default action that the coexistence utility performs at prescribed stages (what to do before CA Access Control installation, after CA Access Control installation, and so on).

How the Services and Controller Plug-In Works

The coexistence utility runs the Services and Controller plug-in to scan the computer for the Windows services management executable at the end of a CA Access Control installation and whenever the utility runs, as follows:

1. Checks if the operating system version is Windows Vista or later

   If the OS is of an earlier Windows version, the plug-in terminates.

2. Creates the following SPECIALPGM resource of type KILL:

   `%windir%\system32\services.exe`

   This is the default action as defined in the response file.

Note: The response file determines the default action that the coexistence utility performs at prescribed stages (what to do before CA Access Control installation, after CA Access Control installation, and so on).

How the Resource Hosting Subsystem Plug-In Works

The CA Access Control installation process runs the Resource Hosting Subsystem plug-in during a CA Access Control installation, and the coexistence utility runs the Resource Hosting Subsystem when a customer executes the utility. The Resource Hosting Subsystem plug-in scans the computer for the Cluster Service Element, as follows:

1. Checks if the operating system is Windows Server 2008 or later.

   If the OS is of an earlier Windows version, the plug-in terminates.

2. Checks if the Cluster Service Element is installed on the computer.

   If the Cluster Service Element is not installed, the plug-in terminates.

3. Creates the following SPECIALPGM resource of type PBF:

   system_drive`:\Windows\Cluster\rhs.exe`

Note: The response file determines the default action that the coexistence utility performs at prescribed stages (what to do before CA Access Control installation, after CA Access Control installation, and so on).

response.ini—Configure the Coexistence Utility

Valid on Windows

The response file instructs the coexistence utility (eACoexist) what actions to perform when it runs. The response file contains a predefined set of actions for every plug-in that the coexistence utility runs. You can edit the response file to change the default plug-in actions.

Note: The response file pathname is specified in the ResponseFile configuration setting of the SeOSD section. This file is *ACInstallDir*\Data\response.ini by default.

This file has the following format:

```
[Section Name]
Act-Stage-#=Action
...
```

Section Name

Defines the name of a section that matches a coexistence plug-in.

The coexistence utility runs the plug-in according to the actions that are defined in this section.

Act-*Stage-#=Action#*

Defines the action you want the plug-in to perform at the prescribed stage.

Stage

Specifies the prescribed stage in which you want the plug-in to perform the action, as follows:

- **BeginInstall**—The plug-in performs the specified action before CA Access Control starts installing.
- **EndInstall**—The plug-in performs the specified action after the CA Access Control installation completes.
- **Utility**—The plug-in performs the specified action when you execute the coexistence utility.
- **BeginUninstall**—The plug-in performs the specified action before CA Access Control starts the uninstall.
- **EndUninstall**—The plug-in performs the specified action after the CA Access Control uninstall completes.

#

Specifies the order in which the plug-in executes actions in this stage.

Action

Specifies a number that defines the action the plug-in should take, as follows:

- **1**—Warn that CA Access Control is not compatible with discovered products.
- **2**—Stop services during CA Access Control installation.
- **3**—Stop services.
- **4**—Start services.
- **5**—Create SPECIALPGM rules.
- **6**—Stop services during CA Access Control uninstall.

Example: Dr. Watson Plug-in Actions

This example displays the default action the Dr. Watson coexistence plug-in performs by default when discovering the Dr. Watson program on the computer.

```
[DrWatson]
Act-EndInstall-0=5
Act-Utility-0=5
```

This section specifies that when the plug-in runs after a CA Access Control installation completes, it should create SPECIALPGM rules for the program. It also specifies that it should do the same when you execute the utility.

eACSigUpdate Utility—Replace STOP Signature File

Valid on Windows

The eACSigUpdate utility replaces the local stack overflow protection (STOP) signature file with a file you updated on another computer.

Note: The eACSigUpdate utility automatically runs when CA Access Control starts, and then at a regular interval, if a signature file broker or a parent Policy Model is defined.

This command has the following format:

`eACSigUpdate` *`hostname target_file`*

hostname

Defines the name of the host computer that has the updated the STOP signature file you want to copy to this computer

Note: For the command to work, you must have administration privileges on the remote host.

target_file

Defines the full path and name of the new signature file. This is the location and name of the signature that is retrieved from the specified host.

eACSyncLockout Utility—Synchronize Account Lockout

Valid on Windows

The eACSyncLockout utility synchronizes an account's lockout with the CA Access Control database. (That is, upon account lockout, the corresponding user's record in the CA Access Control database is suspended.) This utility is effective only when password synchronization is on *and* the user running the utility has the ADMIN property.

This command has the following format:

```
eACSyncLockout -start [-u username] [-p password]

eACSyncLockout -stop|-remove|-debug
```

-p *password*

Defines the user password for the service to be installed and started. If -p is not specified, the utility assumes the user has no password.

-remove

Causes the service to be stopped and uninstalled. (In the next boot of the machine, the service does not appear in the Service Control Manager.)

-start

Causes the service to be installed and started. If -u is not specified, the utility installs and starts the service in the current user's context.

-stop

Stops the service.

-u *user*

Defines the user context for installing and starting the service.

exporttngdb Utility—Migrate Unicenter Security Data

The exporttngdb program migrates the current Unicenter Security data into a local CA Access Control database or PMDB.

Note: On UNIX, two scripts, uni_migrate_master.sh and uni_migrate_node.sh automatically execute this program. Even though both scripts are run on the master machine, the uni_migrate_master.sh script calls it first to migrate the global Unicenter Security data into the Global PMDB. The uni_migrate_node.sh script calls it to migrate the local Unicenter Security data to the local SeOS DB.

This command has the following format:

```
exporttngdb
```

issec Utility—Display CA Access Control Daemon Status

Valid on UNIX

The issec utility displays the status of CA Access Control security daemons. If you do not specify any options, the following information appears:

- The CA Access Control version and installation directory
- The status of the CA Access Control kernel extension
- The status of three major CA Access Control daemons: seosd, agent, and watchdog
- The status of CA Access Control daemons: serevu, selogrd, selogrcd, eacws, ReportAgent, policyfetcher, KBLAudMgr
- The status of the PMDB daemon and its name
- The status of the daemons that have been specified in the [daemons] section of seos.ini

This command has the following format:

```
issec [-b] [-k] [-h]
```

-b

Displays the status and pid of major daemons (seosd, agent, and watchdog).

-k

Checks if CA Access Control kernel extension is loaded.

-h

Displays the help for this utility.

ldap2seos Script—Extract Users from LDAP for Adding into CA Access Control

Valid on UNIX

The ldap2seos utility extracts users from an LDAP database located at the server host and adds them to the CA Access Control database.

Important! CA Access Control lets you use LDAP users directly without importing them if the LDAP user store is used by the operating system, that is, it is an enterprise user store. Consider using this functionality of CA Access Control instead of the ldap2seos utility.

The ldap2seos utility extracts information from an LDAP server about the defined users. The extracted information is automatically used to execute selang commands to add the users to the database. The generated commands are also printed to the standard output and saved automatically to the file named /tmp/ldap2seos.tcl.log.

This utility requires access to a TCL shell environment. The ldap2seos script assumes that the TCL shell path is /usr/local/bin/tclsh. If the TCL shell is placed elsewhere, change the first line in the script.

For the utility to work correctly, CA Access Control must be running. The utility updates the database, so it must be run by a user with the ADMIN privilege. This user must also be authorized in the LDAP database settings to make the search query.

This script has the following format:

`ldap2seos [options]`

-accfld *account-field*

Specifies the LDAP field name containing the user ID for CA Access Control.

If the UNIX user ID is in the LDAP userid field, this option is unnecessary.

If the UNIX user ID is assigned to an LDAP field other than the userid field, specify the LDAP field as *account-field* and the LDAP userid field is ignored.

Note: If the script cannot find the userid, users are not uploaded to the CA Access Control database.

-b *base-entry*

Specifies the base entry, in the LDAP database, from which the users are taken. The entry must be valid inside the LDAP database. If the base entry is omitted, LDAP uses the default base entry to provide the users.

-d *dn*

Specifies an entry name to be used with the *-w* switch to authenticate to LDAP as another user; mostly needed to log into LDAP as admin user.

-f *filename*

Specifies a file to which data retrieved from the LDAP server may be temporarily stored.

-h

Displays help for this utility. The screen contains a listing and explanation of ldap2seos usage and options.

-h *ldap-host*

Specifies the name of the host where the LDAP database is located. The default is the local host.

-l *ldap-dir*

Specifies the directory containing the line command utilities that are assumed to be in the bin subdirectory. The default is /usr/local/ldap.

-p *port*

Specifies the port LDAP uses for connections. The default is port 389.

-u

Identical to -h, displays help. The screen contains a listing and explanation of ldap2seos usage and options.

-w *bindpasswd*

Specifies the user password. To be used with the -d option where authentication is required to access the LDAP database.

Example: Extract User Information

The following command extracts information about users from the LDAP database at host myhost.mysite.com and tries to add them to the CA Access Control database.

```
ldap2seos -h myhost.mysite.com
```

seos2ldap Script—Export CA Access Control Users to LDAP

seos2ldap exports CA Access Control users from the database to an LDAP database located at a server host. It extracts appropriate information about users from the CA Access Control database. It then transmits the information to the selected server's LDAP database. The extracted information is used to generate an LDIF file. Specified users are added to the LDAP database. The responses are saved automatically to the file named /tmp/seos2ldap.tcl.log.

This utility requires access to a TCL shell environment. ldap2seos assumes that the TCL shell path is /usr/local/bin/tclsh. If the TCL shell is placed elsewhere, change the first line in the script.

For the utility to work correctly, CA Access Control must be running. The utility reads from the database, so it must be run by a user with the ADMIN privilege. This user must also be authorized in the LDAP database settings to make changes.

The entry schema, if you elect to use one, for the LDAP database should look like the schema for the Netscape server. If you have changed the Netscape schema, or are using another type of LDAP server, you may need to edit the seos2ldap sample script accordingly.

If a CA Access Control database user already appears in the LDAP database, the user is not added. An error message is produced but the export process continues.

This script has the following format:

```
seos2ldap [options]
```

-b *base-entry*

Specifies the base entry, in the LDAP database, that stores user information. The entry must be valid inside the LDAP database. If the base entry is omitted, LDAP prompts the user to provide it.

-d *dn*

Specifies an entry name to be used with the -*w* switch to authenticate to LDAP as another user. This option is required to log into LDAP as an admin user.

-f *filename*

Specifies a file to which data retrieved from the LDAP server may be temporarily stored.

-h

Displays a help for the utility. The screen contains a listing and explanation of ldap2seos usage and options.

-h *ldap-host*

Specifies the name of the host where the LDAP database is located. The default is the local host.

-l *ldap-dir*

Specifies the directory containing the line command utilities that are assumed to be in the bin subdirectory. The default is /usr/local/ldap.

-noprompt

Cancels base entry prompt. If you did not use the -b *base-entry* flag to specify the base LDAP entry, by default seos2ldap prompts for a base entry. This flag suppresses the prompt.

-p *port*

Defines the port LDAP uses for connections. The default is port 389.

-u

Identical to -h, displays help. The screen contains a listing and explanation of ldap2seos usage and options.

-w *bindpasswd*

Defines the user password. Use this with the -d option where authentication is required to access the LDAP database.

Example: Export User Information

The following command extracts information about users from the CA Access Control database and creates an LDIF file named SeOS_user_dump. The command adds records to the LDAP database at host myhost.mysite.com. You can edit the LDIF file later and update LDAP manually.

```
seos2ldap -h myhost.mysite.com
```

migopts Utility—Translate Unicenter Security Settings

The migopts utility translates current Unicenter security environment settings into the global settings of either a local CA Access Control database or PMDB.

Note: The installation program automatically executes this script when Unicenter Integration is selected. It can, and should, be called manually whenever a new PMDB is created.

This command has the following format:

```
migopts [options]
```

-d *pmdName*

Issues a CA Access Control **hosts** command before running any selang commands to update the imported PMDB (rather than the local CA Access Control database, which is the default).

-f *fileName*

Generates any **selang -c** commands into an executable script file.

-l *logfileName*

Writes log messages to the fully specified file name.

ntimport Utility—Import Windows Users and Groups

Valid on Windows

The ntimport utility extracts Windows users and groups from the Windows operating system database for import into a local database. The utility creates the Windows commands necessary to add users and groups to the local CA Access Control database.

Important. CA Access Control lets use Windows users and groups directly, without needing to import them into the database. Consider using this functionality of CA Access Control instead of the ntimport utility, which was developed before CA Access Control could use Windows users and groups directly.

The generated commands are displayed to the standard output. Use the option -f if you want to create a file to use as input to the selang utility.

This command has the following format:

```
ntimport {-a|{[-u] [-g] [-c]}} [-d] [-U] \
    [-D] [-f filename] [-o owner] [-p pmdb] \
    [-pa pmdb] [-r remote-host] [-v]
```

-a

Performs all actions of the -c, -g, and -u switches.

-c

Generates the selang commands required to join users to their default groups.

-d

Imports users and groups with their domain as prefix.

-D

Retrieves user and group information from the first available domain controller.

-f *filename*

Redirects the output to the specified file.

-g

Generates selang commands required to import groups from Windows to the local database.

-o *owner*

Sets ownership rules for each imported record. Use this flag, to prevent *Administrator* from automatically becoming the owner of all the records. *Owner* is the name of the user or group to be assigned ownership of all records defined by ntimport.

-p *pmdb*

Generates commands for importing user and groups into the AC environment of the pmdb.

-pa *pmdb*

Generates commands for importing user and groups into the AC and native environments of the pmdb.

-pn *pmdb*

Generates commands for importing user and groups into native environment of the pmdb.

-r *remote-host*

Retrieves user and group information from specified remote-host.

-u

Generates the selang commands required to import users from the Windows database to the local database. Names longer than 40 characters are truncated.

-U

Generates the selang commands required to import surrogate rules for users.

-v

Provides the user with progress information. Use this flag to verify the program's progress when there are many users or groups.

policydeploy Utility—Manage Enterprise Policy Deployment

The policydeploy utility manages multiple-rule policies (advanced policy management). It lets you store policy versions on DMS nodes, assign policies to hosts and host groups and unassign these policies, directly deploy or undeploy a stored policy, or upgrade deployed policies to the latest version.

The utility handles several tasks and has the following functions:

Task	Function
Assign or unassign a policy (see page 75)	`policydeploy -assign`
Delete a policy (see page 77)	`policydeploy -delete`
Deploy a policy (see page 78)	`policydeploy -deploy`
Undeploy a policy (see page 78)	`policydeploy -undeploy`
Re-execute a deployment task (see page 79)	`policydeploy -fix`
View deployment scripts (see page 80)	`policydeploy -getrules`
Join or remove a host to a host group (see page 81)	`policydeploy -join`
Migrate a PMD to advanced policy management (see page 82)	`policydeploy -migrate`
Reset policy deployment (see page 85)	`policydeploy -reset`
Restore all policies (see page 85)	`policydeploy -restore`
Store a policy (see page 86)	`policydeploy -store`
Upgrade a policy version (see page 89)	`policydeploy -upgrade`
Downgrade a policy version (see page 89)	`policydeploy -downgrade`

policydeploy -assign Function—Assign or Unassign a Policy

This function assigns or unassigns the specified policy to one or more hosts or host groups.

This function has the following format:

```
policydeploy -assign[-] name -hnode|-ghnode list [-dms list]
```

-assign *name*

Assigns the specified policy to one or more hosts or host groups.

-assign- *name*

Unassigns the specified policy from one or more hosts or host groups.

-dms *list*

(Optional) Specifies a comma-separated list of DMS nodes to use. When you deploy or undeploy a policy, these are the DMS nodes to which the action is reported. When you store a policy, these are the DMS nodes where the policy is stored.

If you do not specify DMS nodes with this option, the utility uses the list of DMS nodes specified in the local CA Access Control database. To specify a list of DMS nodes in the database, you need to issue the following selang command after you create a new DMS using dmsmgr:

```
so dms+(new_dms_name)
```

Note: You need to issue the same command if you did not specify the DMS node during installation, or if you want to replace or add the registered DMS on the endpoint. However, when you specify to create an advanced policy management server during installation, the DMS is added to the database and you do not need to manually run the above command.

-ghnode *list*

Defines a comma-separated list of host groups (GHNODE objects) that you want to assign the policy to.

-hnode *list*

Defines a comma-separated list of hosts (HNODE objects) that you want to assign the policy to.

Example: Assign an IIS 5 Protection Policy

The following example shows you how to assign a policy for securing Internet Information Services (IIS) 5 web servers. We will review the policy and the latest (fourth) version of policy IIS5 and then assign the policy to a host group called IIS5Servers. Policy IIS5 is stored on the crDMS@cr_host.company.com DMS node.

1. Connect to the DMS using selang:

   ```
   hosts crDMS@cr_host.company.com
   ```

   You can now query our DMS via selang.

2. If you're not sure what is the latest finalized version of the policy, issue the following selang command to find all versions of the policy:

   ```
   sr GPOLICY IIS5
   ```

   The selang window lists the properties of the IIS5 policy, including the Final Policy, which is the latest version of the policy that you can assign (finalized).

3. Issue the following selang command to view the policy deployment and undeployment scripts:

   ```
   sr RULESET IIS5#04
   ```

   The selang window displays the IIS5#04 RULESET object, including the deployment and undeployment rules that relate to the fourth version of the IIS5 policy.

4. In a command prompt window, run the policydeploy utility:

   ```
   policydeploy -assign IIS5 -ghnode IIS5Servers
   ```

   This assigns the IIS5 policy to all hosts in the IIS5Servers logical host group, and in turn deploys the fourth version of the IIS5 policy on these hosts.

Example: Unassign an IIS 5 Protection Policy

The following example shows you how to unassign an assigned IIS 5 policy from the web servers that we assigned it to in the previous example.

In a command prompt window, run the policydeploy utility:

```
policydeploy -assign- IIS5 -ghnode IIS5Servers
```

This unassigns the IIS5 policy from all hosts in the IIS5Servers logical host group, and in turn undeploys the version of the IIS5 policy that is deployed on these hosts.

policydeploy -delete Function—Delete a Policy

This function deletes the specified policy or policy version.

Note: You cannot delete a policy or policy version that is assigned to a host or host group, deployed on a host or host group, that has a status of Undeployed with failures, or that has a status on the DMS. Ensure that you undeploy or unassign a policy or policy version from all hosts and host groups before you delete the policy or policy version. In addition, you cannot delete a policy that is a prerequisite for another policy. Remove any dependencies on a policy before you delete it.

This function has the following format:

```
policydeploy -delete name[#xx] [-dms list]
```

-delete *name*[#*xx*]

Deletes the specified policy or policy version.

-dms *list*

(Optional) Specifies a comma-separated list of DMS nodes to use. When you deploy or undeploy a policy, these are the DMS nodes to which the action is reported. When you store a policy, these are the DMS nodes where the policy is stored.

If you do not specify DMS nodes with this option, the utility uses the list of DMS nodes specified in the local CA Access Control database. To specify a list of DMS nodes in the database, you need to issue the following selang command after you create a new DMS using dmsmgr:

```
so dms+(new_dms_name)
```

Note: You need to issue the same command if you did not specify the DMS node during installation, or if you want to replace or add the registered DMS on the endpoint. However, when you specify to create an advanced policy management server during installation, the DMS is added to the database and you do not need to manually run the above command.

Example: Delete an Unassigned IIS 5 Protection Policy

The following example shows you how to delete an unassigned IIS 5 policy from the DMS. In this example, policy IIS5 is not assigned to any hosts or host groups and is is stored on the crDMS@cr_host.company.com DMS node.

To delete the IIS 5 protection policy, open a command prompt window and run the policydeploy utility:

```
policydeploy -delete IIS5
```

Policy IIS5 is deleted from the crDMS@cr_host.company.com DMS node.

Example: Delete an IIS 5 Protection Policy Version

The following example shows you how to delete the unassigned policy version IIS5#05 from the DMS. In this example, policy version IIS5#05 is not assigned to any hosts or host groups and is stored on the crDMS@cr_host.company.com DMS node.

To delete the IIS 5 protection policy version, open a command prompt window and run the policydeploy utility:

```
policydeploy -delete IIS5#05
```

Policy version IIS5#05 is deleted from the crDMS@cr_host.company.com DMS node.

policydeploy -deploy Function—Deploy or Undeploy a Policy

This function deploys and undeploys policies on the specified endpoints, without assigning policies to a host or unassigning policies from a host.

This function has the following format:

```
policydeploy { -deploy name[#xx] | -undeploy name[#xx] } {-nodelist hnode_list | -root dbs} [-dms list]
```

-deploy *name*[*#xx*]

Prompts you for whether you want to directly deploy the specified stored policy version (without assigning the policy to the host) on defined endpoints. To deploy the latest stored version of the policy, omit the policy version number.

-dms *list*

(Optional) Specifies a comma-separated list of DMS nodes to use. When you deploy or undeploy a policy, these are the DMS nodes to which the action is reported. When you store a policy, these are the DMS nodes where the policy is stored.

If you do not specify DMS nodes with this option, the utility uses the list of DMS nodes specified in the local CA Access Control database. To specify a list of DMS nodes in the database, you need to issue the following selang command after you create a new DMS using dmsmgr:

```
so dms+(new_dms_name)
```

Note: You need to issue the same command if you did not specify the DMS node during installation, or if you want to replace or add the registered DMS on the endpoint. However, when you specify to create an advanced policy management server during installation, the DMS is added to the database and you do not need to manually run the above command.

-nodelist *hnode_list*

Defines a comma-separated list of hosts (HNODE objects) that you want to perform the operation for.

-root *dbs*

Defines a comma-separated list of databases where the policy should be deployed or undeployed.

Note: If the root database is a Policy Model parent, the policy is deployed or undeployed throughout its subscribing databases. If the root database is a CA Access Control endpoint, the policy is deployed or undeployed on the specified database only. This option is for backward compatibility with r8 SP1 databases and PMDBs.

-undeploy *name*[*#xx*]

Prompts you for whether you want to directly undeploy the specified policy version *name#xx* (without unassigning the policy) from defined endpoints.

To undeploy the latest stored version of the policy, omit the policy version number.

policydeploy -fix Function—Re-execute Deployment Task

This function fixes the specified deployment task or package and redeploys the task or package.

This function has the following format:

```
policydeploy -fix {-task list | -package list} [-dms list]
```

-dms *list*

(Optional) Specifies a comma-separated list of DMS nodes to use. When you deploy or undeploy a policy, these are the DMS nodes to which the action is reported. When you store a policy, these are the DMS nodes where the policy is stored.

If you do not specify DMS nodes with this option, the utility uses the list of DMS nodes specified in the local CA Access Control database. To specify a list of DMS nodes in the database, you need to issue the following selang command after you create a new DMS using dmsmgr:

```
so dms+(new_dms_name)
```

Note: You need to issue the same command if you did not specify the DMS node during installation, or if you want to replace or add the registered DMS on the endpoint. However, when you specify to create an advanced policy management server during installation, the DMS is added to the database and you do not need to manually run the above command.

-fix

Fixes and redeploys the specified deployment task or package.

-package *list*

Defines a comma-separated list of deployment packages (GDEPLOYMENT).

-task *list*

Defines a comma-separated list of deployment tasks.

policydeploy -getrules Function—View Deployment Scripts

This function lets you view the selang deployment and undeployment scripts for the specified policy version.

```
policydeploy -getrules name[#xx] -ds file1 -uds file2 [-dms list]
```

-dms *list*

(Optional) Specifies a comma-separated list of DMS nodes to use. When you deploy or undeploy a policy, these are the DMS nodes to which the action is reported. When you store a policy, these are the DMS nodes where the policy is stored.

If you do not specify DMS nodes with this option, the utility uses the list of DMS nodes specified in the local CA Access Control database. To specify a list of DMS nodes in the database, you need to issue the following selang command after you create a new DMS using dmsmgr:

```
so dms+(new_dms_name)
```

Note: You need to issue the same command if you did not specify the DMS node during installation, or if you want to replace or add the registered DMS on the endpoint. However, when you specify to create an advanced policy management server during installation, the DMS is added to the database and you do not need to manually run the above command.

-ds *file1*

Specifies the path name of the file containing the deployment rules. These are the commands necessary to construct the policy. When you use the -getrules option, the utility creates this file.

Important! Policy deployment does not support commands that set user passwords. Do not include such commands in your deployment script file. Native selang commands are supported but do not appear in deviation reports.

-getrules *name*[*#xx*]

Retrieves the selang deployment and undeployment scripts for the specified policy version. If you do not specify a policy version, the command applies to the latest policy version.

-uds *file2*

Defines the path name of the file containing the rules required to undeploy the policy. These are the commands necessary to undeploy the policy. When you use the -getrules option, the utility creates this file.

When CA Access Control undeploys a policy, if there is no policy undeployment script stored, CA Access Control calculates the commands required to remove the policy.

Example: View the Deployment Scripts Associated with an IIS 5 Protection Policy

The following example shows you how to view the selang scripts associated with deploying and undeploying a policy for securing Internet Information Services (IIS) 5 web servers. The name of the policy is myPolicy.

To view the selang scripts, run the following command:

```
policydeploy -getrules myPolicy -ds c:\folder\deployRules.txt -uds undeployRules.txt
```

policydeploy -join Function—Join or Remove a Host to a Host Group

This function joins a host to a host group or removes a host from a host group.

This function has the following format:

```
policydeploy -join[-] hnode_name -ghnode name [-dms list]
```

-dms *list*

(Optional) Specifies a comma-separated list of DMS nodes to use. When you deploy or undeploy a policy, these are the DMS nodes to which the action is reported. When you store a policy, these are the DMS nodes where the policy is stored.

If you do not specify DMS nodes with this option, the utility uses the list of DMS nodes specified in the local CA Access Control database. To specify a list of DMS nodes in the database, you need to issue the following selang command after you create a new DMS using dmsmgr:

```
so dms+(new_dms_name)
```

Note: You need to issue the same command if you did not specify the DMS node during installation, or if you want to replace or add the registered DMS on the endpoint. However, when you specify to create an advanced policy management server during installation, the DMS is added to the database and you do not need to manually run the above command.

-ghnode *name*

Defines the name of the host group for the operation you want to perform.

-join *hnode_name*

Adds the specified host to the host group.

-join- *hnode_name*

Removes the specified host from the host group.

policydeploy -migrate Function—Migrate a PMD to Advanced Policy Management

This function migrates a PMD to the advanced policy management environment. When you migrate a PMD to advanced policy management, you create policies from the rules in the PMD, create a host group and hosts in the DMS, and assign the policies to the host group.

This function has the following format:

```
policydeploy -migrate pmdName@hostName [-dms name] [-policydir directory] \
[-exportfilter "class, class..."] [-hgcreate] [-pcreate name] [-addpmdfilter]\
[-unsubs] [-delete] [-auto]
```

pmdName@hostName

Defines the name of the PMD to migrate.

-dms *name*

(Optional) Defines the name of the DMS that the rules in the PMD will be migrated to. If you do not specify the DMS name, the DMS name is retrieved from the CA Access Control database on the local host.

Note: If you do not specify a DMS name and there is more than one DMS name specified in the CA Access Control database on the local host, the rules in the PMD are migrated to all specified DMSs.

-policydir *directory*

(Optional) Defines the directory in which the policy file is stored. If you do not specify a directory, the policy file is stored in your current working directory.

The name of the policy file is *pmdName_hostName*_policy.

-exportfilter "*class*, *class*..."

(Optional) Specifies the CA Access Control classes to export from the PMD database. If you do not specify any classes, all classes in the PMD database are exported.

The following points apply to the -exportfilter parameter:

- If you export rules that modify resources in a particular class, and the class has a corresponding resource group, CA Access Control also exports the rules that modify resources in that resource group.
- If you export rules that modify resources in a particular resource group, CA Access Control also exports the rules that modify the member resource of the resource group.
- If you export rules that modify resources in a particular class and that class has a PACL, CA Access Control also exports the rules that modify resources in the PROGRAM class.
- If you export rules that modify resources in a particular class and that class has a CALACL, CA Access Control also exports the rules that modify resources in the CALENDAR class.
- If you export rules that modify resources in a particular class, and one of the resources in that class is a member of a CONTAINER resource group, CA Access Control exports the rules that modify resources in the CONTAINER class and the rules that modify the resources that are members of each CONTAINER resource group.

-hgcreate

(Optional) Creates a host group (GHNODE object) on the DMS that corresponds to *pmdName*, creates hosts (HNODE objects) on the DMS that correspond to endpoint subscribers of *pmdName*, and joins the hosts to the host group.

-pcreate *name*

(Optional) Creates a POLICY object on the DMS that contains the rules in the policy file that was exported from *pmdName*, and assigns the POLICY object to the host group on the DMS that corresponds to *pmdName*. If you specify *name*, the created POLICY object is named *name*_POLICY#01; if you do not specify name, the created POLICY object is named *pmdName*_POLICY#01.

-addpmdfilter

(Optional) Applies a filter file to *pmdName.* The filter file is named filter.flt and is located in the same directory as *pmdName*.

Note: You use the filter file to create a password PMD. The filter file lets only user password commands be sent to the subscribers of *pmdName*.

-unsubs

(Optional) Unsubscribes endpoint subscribers from *pmdName*.

-delete

(Optional) Deletes *pmdName* after the policydeploy -migrate function has finished executing.

-auto

(Optional) Specifies to execute both the -hgcreate and -pcreate options. This option does the following:

- Exports the rules in pmdName
- Creates a host group (GHNODE object) on the DMS that corresponds to *pmdName*
- Creates hosts (HNODE objects) on the DMS that correspond to endpoint subscribers of *pmdName*
- Joins the hosts to the host group
- Creates a POLICY object on the DMS that contains the rules in the policy file that was exported from *pmdName*
- Assigns the POLICY object to the host group on the DMS that corresponds to *pmdName*

Example: Migrate Rules and Create a Host Group

This example migrates the rules from Master PMD on host A to DMS__ on host B, saves the policy file to the C:\Data\policies_MasterPMD_hostA directory, creates a host group named MasterPMD on DMS__, creates hosts on DMS__ that correspond to the endpoint subscribers of Master PMD, and joins the hosts to the MasterPMD host group:

```
policydeploy -migrate MasterPMD@hostA -dms DMS__@hostB -policydir
"C:\Data\policies_MasterPMD_hostA" -hgcreate
```

policydeploy -reset Function—Reset Policy Deployment

This function resets policy deployment on the endpoint. CA Access Control undeploys all the effective policies on the endpoint, deletes all advanced policy management properties, and resets host status.

This function has the following format:

```
policydeploy -reset hnode_name [-dms list]
```

-dms *list*

(Optional) Specifies a comma-separated list of DMS nodes to use. When you deploy or undeploy a policy, these are the DMS nodes to which the action is reported. When you store a policy, these are the DMS nodes where the policy is stored.

If you do not specify DMS nodes with this option, the utility uses the list of DMS nodes specified in the local CA Access Control database. To specify a list of DMS nodes in the database, you need to issue the following selang command after you create a new DMS using dmsmgr:

```
so dms+(new_dms_name)
```

Note: You need to issue the same command if you did not specify the DMS node during installation, or if you want to replace or add the registered DMS on the endpoint. However, when you specify to create an advanced policy management server during installation, the DMS is added to the database and you do not need to manually run the above command.

-reset *hnode_name*

Resets policy deployment on the specified endpoint.

policydeploy -restore Function—Restore All Policies

This function undeploys any policies on the specified host, then restores (directly redeploys) all the policies that should be deployed (assigned or directly deployed) on the host by resending all the deployment tasks to the host for execution.

Important! If the host has some policies already applied, the restore will fail because it does not reset the host status before executing. Use the policydeploy -reset function instead.

This function has the following format:

```
policydeploy -restore hnode_name [-dms list]
```

-dms *list*

(Optional) Specifies a comma-separated list of DMS nodes to use. When you deploy or undeploy a policy, these are the DMS nodes to which the action is reported. When you store a policy, these are the DMS nodes where the policy is stored.

If you do not specify DMS nodes with this option, the utility uses the list of DMS nodes specified in the local CA Access Control database. To specify a list of DMS nodes in the database, you need to issue the following selang command after you create a new DMS using dmsmgr:

```
so dms+(new_dms_name)
```

Note: You need to issue the same command if you did not specify the DMS node during installation, or if you want to replace or add the registered DMS on the endpoint. However, when you specify to create an advanced policy management server during installation, the DMS is added to the database and you do not need to manually run the above command.

-restore *hnode_name*

Restores (directly redeploys) all the policies that should be deployed on the specified host.

policydeploy -store Function—Store a Policy

This function stores the specified policy on the DMS nodes specified by the command or in the local CA Access Control database. Unless you use the -silent option, you need to confirm this action at the prompt.

If no previous version of the specified policy is stored on the DMS, version 1 of the policy is created (*name*#01). If a previous version of this policy exists, a new version of the policy is created (*name#last_version+1*). The policy version you store is automatically finalized. When you need to update a policy, you must store a new version of the policy that contains the required modified policy deployment and undeployment rules.

This function has the following format:

```
policydeploy -store name -ds file1 [-uds file2] [-dms list] [-desc description]
[-prereq list] [-silent]
```

-desc *description*

(Optional) Defines the business description for the policy.

-dms *list*

(Optional) Specifies a comma-separated list of DMS nodes to use. When you deploy or undeploy a policy, these are the DMS nodes to which the action is reported. When you store a policy, these are the DMS nodes where the policy is stored.

If you do not specify DMS nodes with this option, the utility uses the list of DMS nodes specified in the local CA Access Control database. To specify a list of DMS nodes in the database, you need to issue the following selang command after you create a new DMS using dmsmgr:

```
so dms+(new_dms_name)
```

Note: You need to issue the same command if you did not specify the DMS node during installation, or if you want to replace or add the registered DMS on the endpoint. However, when you specify to create an advanced policy management server during installation, the DMS is added to the database and you do not need to manually run the above command.

-ds *file1*

Specifies the path name of the file containing the deployment rules. These are the commands necessary to construct the policy. When you use the -getrules option, the utility creates this file.

Important! Policy deployment does not support commands that set user passwords. Do not include such commands in your deployment script file. Native selang commands are supported but do not appear in deviation reports.

-prereq *list*

(Optional) Defines a comma-separated list of policies that must be deployed before you can deploy this policy.

Important! If a prerequisite policy is not deployed when you try to deploy a dependent policy, the deployment task's status is changed to *Pending Prerequisite* and the deployment resumes when all prerequisite policies are deployed. Similarly, if you try to undeploy a policy that is a prerequisite to another deployed policy, the deployment task's status is changed to *Pending Dependents* and the deployment resumes when all dependent policies are undeployed.

-silent

(Optional) Suppress the confirmation prompt for the requested action.

-store *name*

Stores the specified policy on the specified DMS nodes or in the local CA Access Control database.

Note: Policy names cannot include the # (hash) character which is reserved for denoting policy version numbers and is added automatically.

-uds *file2*

Defines the path name of the file containing the rules required to undeploy the policy. These are the commands necessary to undeploy the policy. When you use the -getrules option, the utility creates this file.

When CA Access Control undeploys a policy, if there is no policy undeployment script stored, CA Access Control calculates the commands required to remove the policy.

Example: Store an IIS 5 Protection Policy

The following example shows you how to store a policy for securing Internet Information Services (IIS) 5 web servers. This is the first time we store this policy on the DMS.

Note: The selang commands in this example are for resources on a Windows operating system but the same procedure also applies on UNIX.

1. Save a file named IIS5.selang with the following IIS script:

```
# IIS5 deployment script
eu inet_pers owner(nobody)
er FILE c:\InetPub\wwwroot\* defaccess(none) owner(nobody)
authorize FILE c:\InetPub\wwwroot\* uid(inet_pers) access(all)
er FILE c:\InetPub\wwwroot\scripts defaccess(none) owner(nobody)
er FILE *.asp defaccess(none) owner(nobody)
authorize FILE *.asp uid(inet_pers) via(pgm(inetinfo.exe)) access(read, execute)
```

   These are the commands necessary to deploy an IIS 5 protection policy.

2. Save a file named IIS5_rm.selang with the following script:

```
# IIS5 undeployment script
ru inet_pers
rr FILE c:\InetPub\wwwroot\*
rr FILE c:\InetPub\wwwroot\scripts
rr FILE *.asp
```

   These are the commands necessary to undeploy the IIS 5 protection policy we created in Step 1.

3. Open a command prompt window and run the policydeploy utility:

   ```
   policydeploy -store IIS5 -ds IIS5.selang -uds IIS5_rm.selang -desc "IIS5 web server security policy" -silent
   ```

   This stores on the DMS the policy IIS5 (GPOLICY object) and the first version of the policy (IIS5#01 POLICY object) with the scripts defined in IIS5.selang and IIS5_rm.selang.

policydeploy -upgrade Function—Upgrade or Downgrade a Policy Version

This function upgrades a policy to its latest finalized version on the defined hosts, or downgrades a policy to a specified policy version on the defined hosts.

This function has the following format:

```
policydeploy {-upgrade name | -downgrade name#xx} [-nodelist hnode_list|-ghnode name] [-list] [-dms name]
```

-dms *list*

(Optional) Specifies a comma-separated list of DMS nodes to use. When you deploy or undeploy a policy, these are the DMS nodes to which the action is reported. When you store a policy, these are the DMS nodes where the policy is stored.

If you do not specify DMS nodes with this option, the utility uses the list of DMS nodes specified in the local CA Access Control database. To specify a list of DMS nodes in the database, you need to issue the following selang command after you create a new DMS using dmsmgr:

```
so dms+(new_dms_name)
```

Note: You need to issue the same command if you did not specify the DMS node during installation, or if you want to replace or add the registered DMS on the endpoint. However, when you specify to create an advanced policy management server during installation, the DMS is added to the database and you do not need to manually run the above command.

-downgrade *name#xx*

Downgrades a policy to the specified policy version on the defined hosts.

-ghnode *name*

Defines the name of the host group for the operation you want to perform.

-list

(Optional) Lists the hosts that have a version of the specified policy deployed, that is not the version specified. If you use -upgrade the implicitly specified version is the latest available.

-nodelist *hnode_list*

Defines a comma-separated list of hosts (HNODE objects) that you want to perform the operation for.

-upgrade *name*

Upgrades the specified policy to its latest finalized version on the defined hosts.

Example: Upgrade an IIS 5 Protection Policy

The following example shows you how to upgrade a policy. We will first review the deployment to see which hosts do not have the latest version of this policy deployed.

1. In a command prompt window, run the policydeploy utility:

   ```
   policydeploy -upgrade IIS5 -list
   ```

   This lists the hosts that have an older version of the IIS5 policy deployed.

2. Upgrade all of these hosts to the latest version of the policy:

   ```
   policydeploy -upgrade IIS5
   ```

Example: Downgrade an IIS 5 Protection Policy

The following example shows you how to downgrade a policy. We will first review the deployment to see which hosts have a deployed policy that has earlier versions.

1. In a command prompt window, run the policydeploy utility:

   ```
   policydeploy -downgrade IIS5#3 -list
   ```

   This lists the hosts that have a version of the IIS5 policy deployed that is later than version 3.

2. Downgrade all of these hosts to the third version of the policy:

   ```
   policydeploy -downgrade IIS5#3
   ```

pwextractor Utility—Extract Privileged Account Passwords

The pwextractor utility extracts privileged account passwords from the database. You can use pwextractor if you want to back up privileged account passwords, or if Privileged User Password Management is unavailable and you cannot check out privileged accounts.

To use pwextractor, you must:

- Have access to the database tables
- Know the user name and password for the account that Privileged User Password Management uses to access the database

  Note: You provide these credentials when you install the Enterprise Management Server.

If you use a Microsoft SQL Server database and the database authentication mode is Windows Authentication, when you use pwextractor you must:

- Verify that the sqljdbc_auth.dll file is located in the JAVA_HOME\bin directory
- Use the pwextractor -url format
- Specify integratedSecurity=true; in the JDBC URL string

Note: You can use the pwextractor -url format only when you install the Enterprise Management Server on a Windows computer and use a Microsoft SQL Server database. For more information about the sqljdbc_auth.dll file, see the Microsoft SQL Server documentation.

pwextractor is located in the following directory:

ACServerInstallDir/IAM Suite/Access Control/tools/pwextractor

This command has the following format:

```
pwextractor -h hostname [-r port] -d {database | schema} -t {mssql | oracle} -l login
-p password -f filename [-k key_file]
```

This command has the following format for JDBC databases. This format is valid only when you install the Enterprise Management Server on a Windows computer and use a Microsoft SQL Server database:

```
pwextractor -url url -f filename [-k key_file]
```

-h *hostname*

Defines the name of the database host.

-r *port*

Defines the port number on which the database communicates.

-d {*database* | *schema*}

Defines the following:

- (MS SQL) Defines the database name.
- (Oracle) Defines the schema name.

-t {mssql | oracle}

Specifies the database type.

Values: mssql, oracle

-l *login*

Defines the user name for the account that Privileged User Password Management uses to access the database.

-p *password*

Defines the password for the account that Privileged User Password Management uses to access the database.

-f *filename*

Defines the directory path and file name for the output file. If you specify an existing file, pwextractor replaces the existing file with the new output.

-k *key_file*

Defines the full path and name for the encryption file that was used to encrypt the passwords.

-url *url*

Defines the JDBC URL string that you use to access the database.

Format: jdbc:sqlserver://*servername*:*port*[;*property*=*value*]

Example:
jdbc:sqlserver://localhost:1433;selectMethod=cursor;DatabaseName=mydb;user=sa;password=mypwd;

Example: Extract Privileged User Password Management Passwords from a Microsoft SQL Server Database

The following examples extract the Privileged User Password Management passwords from a Microsoft SQL Server database named mydb and located on host myhost.example.com. The Enterprise Management Server is located on a Windows computer and the encryption file is located at C:\FIPSkey.dat. pwextractor writes the output to the C:\accounts.txt file.

- This example extracts the passwords when the database authentication mode is SQL Server Authentication:

  ```
  pwextractor.bat -h myhost.example.com –r 1433 -d mydb -t mssql -l sa -p mypwd -f
  C:\accounts.txt -k "C:\FIPSkey.dat"
  ```

- This example extracts the passwords when the database authentication mode is Windows Authentication:

  ```
  pwextractor.bat -url
  jdbc:sqlserver://myhost.example.com:1433;selectMethod=cursor;DatabaseName=myd
  b;user=sa;password=mypwd;integratedSecurity=true; -f C:\accounts.txt -k
  "C:\FIPSkey.dat"
  ```

ReportAgent Utility—Send Report Snapshots and Audit Events

The ReportAgent sends report snapshots and audit events to the Distribution Server for inclusion in CA Access Control, UNIX Authentication Broker, and CA Enterprise Log Manager reports.

You must configure an endpoint for reporting before you run the ReportAgent. When you configure an endpoint for reporting, you specify the Distribution Server with which the ReportAgent communicates and the schedule at which it runs. After you configure an endpoint for reporting, the ReportAgent runs as a daemon or service and sends snapshots at the scheduled times. However, if you want to immediately send report snapshots or audit events to the Distribution Server, you can run the ReportAgent on demand.

Note: For more information about how to configure an endpoint for reporting, see the *Implementation Guide*. You can also use the report_agent.sh script to configure, start, and stop the ReportAgent on UNIX computers.

On UNIX computers, you run the ReportAgent utility from the *ACSharedDir*/bin directory on a UNIX computer, where *ACSharedDir* is the default directory /opt/CA/AccessControlShared. You may also need to set the library path environment variable.

This command has the following syntax:

```
ReportAgent -debug {0 | 1 | 2} -task {0 | 1 | 2 | 3 | 4} [-now]
ReportAgent -report snapshot
```

-debug {0 | 1 | 2}

Specifies to run the ReportAgent in debug mode. The ReportAgent service or daemon must be stopped to use this option.

Limits: 0—Prints debug information to the console.

1—Prints debug information to the log file.

2—Does not print debug information (no output).

-task {0 | 1 | 2 | 3 | 4}

Specifies the information that the ReportAgent sends to the Distribution Server.

Limits: 0—Sends a snapshot of the CA Access Control database and any local PMDBs to the queue/snapshots queue on the Distribution Server.

1—Sends endpoint audit events to the queue/audit queue on the Distribution Server.

2—(UNIX) Sends a snapshot of the UNIX Authentication Broker database to the ac_endpoint_to_server queue on the Distribution Server.

3—(UNIX) Sends UNIX Authentication Broker audit events to the queue/audit queue on the Distribution Server.

4—(UNIX) Sends keyboard logger audit events to the queue/audit queue on the Distribution Server.

-now

Specifies to immediately run the ReportAgent.

If you do not specify this option, the ReportAgent runs at the next scheduled time.

-report snapshot

Specifies to immediately send a snapshot of the CA Access Control database and any local PMDBs to the queue/snapshots queue on the Distribution Server. The ReportAgent service or daemon must be running to use this option.

Example: View ReportAgent Debug Information

The following example sets the library path environment variable on a Linux computer, then specifies to immediately run the ReportAgent in debug mode, print debug information to the console, and send audit events to the Distribution Server:

```
LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/opt/CA/AccessControlShared/lib
export LD_LIBRARY_PATH
cd /opt/CA/AccessControlShared/bin
./ReportAgent -debug 0 -task 1 -now
```

ReportAgent Log Files

The following table lists the log files to which the ReportAgent writes debug information when you run the ReportAgent -debug 1 command. In this table, *ACSharedDir* is the default directory /opt/CA/AccessControlShared and *ACInstallDir* is the directory in which you installed CA Access Control:

ReportAgent Option	UNIX Log File	Windows Log File
-task 0	*ACSharedDir*/log/ac2xml.log	*ACInstallDir*\log\ac2xml.log
-task 1	*ACSharedDir*/log/ac2elm.log	*ACInstallDir*\log\ac2elm.log
-task 2	*ACSharedDir*/log/unab2xml.log	-
-task 3	*ACSharedDir*/log/unab2elm.log	-
-task 4	*ACSharedDir*/log/kbl2elm.log	-

report_agent.sh Script—Configure the Report Agent

Valid on UNIX

The report_agent.sh script lets you configure the Report Agent daemon after installation. Use the report_agent.sh script if you need to change the Report Agent configuration settings you set when you installed CA Access Control.

The report_agent.sh script is located in *ACSharedDir*/lbin. By default, this directory is /opt/CA/AccessControlShared/lbin.

This command has the following format:

```
report_agent.sh start
report_agent.sh stop
report_agent.sh config -server hostname [-proto {ssl|tcp}] [-port port_number]
[-rqueue queue_name] \
[-schedule <time@day[,day2,...]>] [-audit] [-bak] [-silent]
```

config

Specifies that the remaining parameters configure the Report Agent daemon.

start

Starts the Report Agent.

stop

Stops the Report Agent.

-server *hostname*

Defines the name of the Distribution Server host. Combined with the input from the -port option, the script constructs the Distribution Server URL and sets the report_server configuration setting in the ReportAgent section.

-audit

Specifies whether you want to send endpoint audit data to the Distribution Server. This sets the reportagent_enabled configuration setting in the ReportAgent section.

-bak

Specifies to keep timestamped backups of audit files. This sets the logmgr section configuration setting Backup_Date to yes and audit_max_files to 50.

-port *port_number*

Defines the port number to use for communication with the Distribution Server. Combined with the input from the -server option, the script constructs the Distribution Server URL and sets the report_server configuration setting in the ReportAgent section.

-proto

Specifies the connection protocol (TCP or SSL). This sets the use_ssl configuration setting in the ReportAgent section

-rqueue *queue_name*

Defines the name of the queue to which the Report Agent sends snapshots of the local database and any PMDBs. This sets the send_queue configuration setting in the ReportAgent section.

-schedule <*time@day*[*,day2,...*]>

Defines when to generate reports and when to send the reports to the Distribution Server.

-silent

Specifies not to ask for confirmation.

Example: Configure the Report Agent

This example sets the Report Agent to send database snapshots to a Distribution Server on rscomp.com using port 7243 over SSL and a queue named queue/snapshots. It also enables sending audit data to the Distribution Server and sets backup settings for the audit log file:

```
report_agent.sh config -server rscomp.com -proto ssl -port 7243 -rqueue
queue/snapshots -audit
```

Once you configure Report Agent, you should update the +reportagent user with a correct password (shared secret) that the Distribution Server expects. To do this, enter the following:

```
eu +reportagent epassword(Shared_Secret) nonnative
```

seaudit Utility—Display Audit Log Records

The seaudit utility displays the records in the CA Access Control audit log file. To execute the seaudit utility on Windows, you must have the AUDITOR attribute. To execute the seaudit utility on UNIX, you must belong to the audir_group in seos.ini. When displaying audit records that include passwords, seaudit protects password identity by substituting a series of asterisks (***) in place of the password text.

Note: You can use string matching in the command switches and options. Some UNIX shells automatically expand mask arguments; therefore, when invoking seaudit from such a shell, you should prevent the masks from being handled by the shell by typing a backslash (\) before an asterisk or question mark.

Note: The seaudit utility displays trace records by user name, not by user ID.

This command has the following format:

```
seaudit switch [options]
```

switch

Defines the mode of operation for the utility. Can be *one* of the following:

-a | -all

Displays all records except user trace records sent to the audit log by the tracing facility.

Note: Connected TCP records, which are available for UNIX, are also not displayed. You need to also specify the -c option to display these records.

-h | -help

Displays the help for this utility.

{-i | -inet} ***host service***

Displays the INET audit records of the TCP requests received from the specified hosts for the specified services. Both *host* and *service* are masks that identify the set of hosts and services that seaudit searches for.

On UNIX, to list TCP records with the network ID (port number) to which connection was made, add the -c flag. For example:

```
seaudit -i -c myhost telnet
```

{-l | -login} ***user1, user2, ... terminal***

Displays the LOGIN records for the comma-separated specified users, on the specified terminal.

Both *user* and *terminal* are masks.

On UNIX, this also lists records created by serevu when it enables and disables users, and records created by the authorization daemon when an invalid password is entered.

{-r | -resource} ***class resource user1, user2, ...***

Displays the general resources audit of the specified class on the specified resource for the specified comma-separated users.

- *class* is a mask that identifies the class to which the accessed resource belongs.
- *resource* is a mask that identifies the names of the resources that were accessed.
- *user* is a mask of the name of the user who accessed the resource.

-s | -start

Displays the CA Access Control startup and shutdown messages.

-St | -Stat ***message_number***

(UNIX only). Displays a description of the watchdog message number.

-t | -table

Displays the table of log codes.

-tr

Displays trace records of all the users whose activities are being traced.

Note: Trace records display the login session ID column by default. If you do not want to display this column, use th -format option.

-trr ***resource***

Displays the trace records of the specified resource.

-tru {*uid1*|*user1*}, {*uid1*|*user2*}, ...

Displays the trace records of the users with the specified numeric uids or user names.

-u *command class record user*

Displays database update audit records:

- *command* is a mask identifying the set of selang commands to search for.
- *class* is a mask identifying the classes to search for.
- *record* is a mask identifying the records to search for.
- *user* is a mask identifying the users who executed the commands.

-w

Displays the watchdog audit records.

options

Defines optional modifiers that change the way the utility displays its information. Can be one or more of the following:

-c

(UNIX only). Displays *connected* INET records. These are records generated for session ID tracking, which list the port number of successful TCP connections.

For example, a user (user1) opens a Telnet session from comp1 to comp2, both with CA Access Control installed. CA Access Control on comp2 can be configured (logconnected configuration setting) to send acknowledgement to comp1 with the credentials of the user who logged in through the Telnet session (this may be a user other than user1). When comp1 receives this acknowledgement, it creates a TCP-CONNECTED record (a session establishment record) that can then be displayed using the -c option.

-detail

Displays detailed information about each record.

-delim *delimiter*

Defines the delimiter to use before the first field and between the remaining fields. For example, the following command makes fields appear in quotation marks separated by a comma:

seaudit -a -delim \",\"

-delim2 *delimiter*

Same as the -delim option, except that the delimiter does not appear before the first field.

-delim3 *delimiter*

Same as the -delim option, except that it includes a delimiter between day, month, and year.

-delim4 *delimiter*

Same as the -delim2 option.

-ed *date*

Specifies the end date. Records logged after this date are *not* displayed.

You can specify *date* in one of two ways:

- Using the format *dd-mm-yyyy*.
- Using the string *today* to set the date as today.

You can also use the string *today* followed by - (minus) and a number. This defines the date as the specified number of days before today. For example, *today-3* means that the date is three days ago.

-et *time*

Specifies the end time. Records logged after this time are *not* displayed.

You can specify *time* in one of two ways:

- Using the 24-hour format *hh:mm*
- Using the string *now* to set the time as now.

  You can also use the string *now* followed by - (minus) and a number. This defines the time as the specified number of minutes before now. For example, now-*60* means that the time is sixty minutes (one hour) ago. To delineate a time frame within a particular day, use this option in conjunction with -sd, -ed or both.

Note: The *now* string is valid for the present day's time. For example, if the present time is 130 am, you specify *now-89*. If you specify *now-90*, then no records appear.

-f | -failure

Specifies *not* to display access failures.

{-fn | -file} *fileName*

Specifies the name of the audit log file to be searched.

-format *release*

Specifies that the output format looks like it did for CA Access Control *release*.

release—Defines the release number. Valid values are:

- **80sp1**—The output in r8 SP1 did not include the effective UID column that exists in newer releases.
- **12**—The output in r12.0 did not include the ability to display password change records. For trace records, the output in r12.0 also did not include login session ID information.

-g | -grant

Specifies *not* to display successful (granted) accesses.

-gn | -grantnotify

Specifies *not* to display successful (granted) accesses, except for notify records.

-kbl -a -sid *sid* {-rp | -pr | -cmd | -exe | -disp}

(UNIX only) Specifies to display the content of the keyboard logging audit file (kbl.audit).

-a

Displays all recorded sessions in the audit file.

-sid *sid*

Specifies the keyboard logging session ID.

-rp

Replays the entire keyboard logging session.

-pr

Displays the entire keyboard logging session, excluding control characters.

-cmd

(UNIX Only) Displays the commands that the user entered during the command line logging session.

-exe

Displays EXECARGS details of commands that the user executed in the shell.

-disp

Specifies to display the recorded session time.

Note: You can run the command in the following shells: bash, tcsh, csh, ksh, jsh, rsh, ash, zsh

-logout

(UNIX only) Specifies *not* to display logout records.

-millennium

(UNIX only) Specifies that years should be displayed with four digits instead of two.

-n | -netaddr

Specifies that Internet addresses should be displayed instead of host names in TCP/IP records.

-notify

Specifies *not* to display NOTIFY audit records.

{-o | -origin} *host*

Specifies that only records originating from the specified host should be displayed.

This option is only applicable when browsing records from a consolidated audit file created by the selogrcd log-routing collection daemon.

-pwa

(UNIX only) Specifies *not* to display password attempt records.

-sd *date*

Specifies the start date. Records logged prior to this date are *not* displayed.

You can specify *date* in one of two ways:

- Using the format *dd-mm-yyyy*.
- Using the string *today* to set the date as today.

You can also use the string *today* followed by - (minus) and a number. This defines the date as the specified number of days before today. For example, *today-3* means that the date is three days ago.

sessionid

Specifies to show a column that contains user login session ID information. This column is hidden by default.

Note: This option is valid only for endpoints with r12.0 SP1 and above.

-st *time*

Specifies the start time. Records logged prior to this time are *not* displayed.

You can specify *time* in one of two ways:

- Using the 24-hour format *hh:mm*
- Using the string *now* to set the time as now.

You can also use the string *now* followed by - (minus) and a number. This defines the time as the specified number of minutes before now. For example, now-*60* means that the time is sixty minutes (one hour) ago. To delineate a time frame within a particular day, use this option in conjunction with -sd, -ed or both.

Note: The *now* string is valid for the present day's time. For example, if the present time is 130 am, you specify *now-89*. If you specify *now-90*, then no records appear.

-v | -servnum

Specifies that port numbers are displayed instead of service names.

-warn

Specifies *not* to display warning records.

Examples

- To list all audit records since 3 January 2004, use the following command:

```
seaudit -a -sd 04-Jan-2004
```

- To list the failed logins of the user root from any terminal on 3 January 2004, use the following command:

```
seaudit -sd 04-Jan-2004 -ed 04-Jan-2004 -l root * -g
```

- To list all accesses of user John to every resource of class FILE, use the following *command:*

```
seaudit -r FILE * John
```

- To list all audit records that were logged between 17:00 (the first day) and 08:00 (the following day), for all dates, use the following command:

```
seaudit -a -st 17:00 -et 08:00
```

- To list all audit records that were logged between 08:00 and 17:00, use the following command:

```
seaudit -a -st 08:00 -et 17:00
```

- To list all warning records for logins and resource accesses for a single user, use the following command:

```
seaudit -login * * -resource * * * -grant -failure -logout -pwa
```

- To list all login records for two users, use the following command:

```
seaudit -login "user1, user2"
```

- To list all audit records from yesterday, use the following command:

```
seaudit -a -sd today-1 -ed today-1
```

- To list all the audit records in the kbl.audit log file, use the following command:

```
seaudit -kbl
```

- To replay a user session, use the following command:

```
seaudit -kbl -sid 22316 -rp
```

- To display all the commands a user entered in a session, use the following command:

```
seaudit -kbl -sid 22316 -cmd
```

- To list all audit records that trace the activity of a single user with UID 244 attempting to access files, use the following command:

  ```
  seaudit -tru 244 -trr FILE
  ```

- To list all audit records that trace the activity of two users, use the following command:

  ```
  seaudit -tru "user1, 244"
  ```

More information:

How To Identify the Event Type of an Audit Record (see page 543)
Audit Event Types (see page 545)

sebuildla Utility—Create a Lookaside Database

Valid on UNIX

The sebuildla utility creates a lookaside database for use by the CA Access Control daemon, seosd. The seosd daemon uses the database to translate UNIX UIDs to user names, GIDs to group names, host IP addresses to host names, and service ports to port names. The database contains only the number to name translation. sebuildla also lets you add information from the LDAP Directory Information Tree (DIT) to the user lookaside database.

Important! To set up sebuildla and the required LDAP configuration settings you must to be familiar with LDAP and be able to execute the ldapsearch command. We recommend that you read the man pages for ldap(1), ldapsearch(1) and the information about setting up in the documentation for your LDAP client. Also, before you use sebuildla to build the lookaside databases, specify the full path of the lookaside database, in the lookaside_path configuration setting.

The first time you build the lookaside database, use the following command:

```
sebuildla -a
```

This creates *all* of its components. Single files of the database can be updated later by using the relevant switches.

If you installed CA Access Control on a NIS, NIS+, or DNS server, you should place calls to the sebuildla utility in the related makefiles.

Note: By default, the lookaside database files (groupdb.la, hostdb.la, servdb.la, and userdb.la) are protected against all user access other than access with the sebuildla program.

The sebuildla utility scans the resolution mechanisms in the system, such as /etc files and NIS, to build the lookaside databases.

- sebuildla reads /etc/resolv.conf to get the domain name used.

  Note: For CA Access Control to resolve host names to fully qualified names, the resolv.conf file must have either the domain or search configuration option defined. For more information about the resolv.conf file, see the man pages for this file.

- sebuildla uses the system resolution option to create the lookasaide database. (This is usually the net caching daemon.)
- CA Access Control uses the /etc/nsswitch.conf file (for the net caching daemon or any other system resolution option) to decide where to retrieve data from.

For example, if the /etc/nsswitch.conf file contains the following line for hosts, information is retrieved from the local machine's files first (/etc/hosts); it then retrieves information from the DNS and then the NIS:

```
hosts:      files dns nis
```

If the file contains the following line instead, information is retrieved only from your local machine's files. The look aside database will contain only the hosts that are in /etc/hosts:

```
hosts:      files
```

Note: If a host has a fully qualified name, sebuildla uses it.

Variations in machine configuration may cause instances where sebuildla does not list all the names of a local environment. In this case, you can use sebuildla to load all the required entries from a list file. To do this, create a list file with each object name on a separate line. The utility reads this list file and ensures that all the objects in the list file are added to the relevant lookaside database if necessary. sebuildla ignores duplicate objects.

The following table lists the files sebuildla uses to build each lookaside database.

Objects in	Are added to the
ACInstallDir/ladb/userlist	users lookaside database
ACInstallDir/ladb/grouplist	groups lookaside database
ACInstallDir/ladb/hostlist	hosts lookaside database
ACInstallDir/ladb/servlist	services lookaside database

In the format of the files in the *ACInstallDir*/ladb directory:

- sebuildla ignores empty lines or lines that begin with an exclamation point (!), number sign (#), or a semicolon (;).
- Other lines represent entries that sebuildla must add to the appropriate lookaside database, if the entry can be resolved.
- The user, group, host, or service name must start in the first position of the line.

You can use dbmgr -dump -r to create the list files. For example, to create a list of the hosts defined in class HOST in the local database, enter:

```
dbmgr -dump -r l HOST > /opt/CA/AccessControl//ladb/hostlist
```

The -l switch makes a single request from DNS for a list of all hosts in the default domain, instead of querying the DNS server for the FQDN of each host entry as it is obtained. The fast load option is effectual only if DNS is installed. Only host names in the default domain are made fully qualified. Fully qualified names are left as such. Host names scanned from the system mechanism that are not fully qualified, and are not found in the default domain, are left unqualified. Host names loaded from the hostlist file that are not fully qualified are discarded.

This command has the following format:

```
sebuildla switch [options]
```

switch

Specifies the mode of operation for the utility. Can be *one* of the following:

-a

Creates *all* the lookaside database files.

-e

Creates a hosts lookaside database file excluding the DNS.

-g

Creates a groups lookaside database file.

-h

Creates a hosts lookaside database file with the DNS.

-help

Displays the help for this utility.

-n

Collects information from an LDAP Directory Information Tree (DIT) and appends it to the users lookaside database it creates from the primary user data source (-u switch). You can only use this switch in conjunction with the -u switch or the -a switch so it is most useful when the LDAP DIT provides additional user data and is not used as the system's naming service.

Before you use this switch, follow these steps:

a. Set the following seos.ini file tokens for CA Access Control to find the LDAP service: ldap_base, ldap_hostname, and ldap_userdn.

b. Run the seldapcred utility to store the encrypted LDAP password.

c. (Optional) Set the ldap_port and ldap_timeout tokens for your environment.

   The time it takes to retrieve information from the LDAP service depends on how fast the LDAP service is, and how much user data is stored in the DIT. You can adjust the ldap_timeout token in the [seos] section of the seos.ini file to account for these aspects.

d. (Optional) If you are using a non-standard schema, set the ldap_uid_attr, ldap_uidNumber_attr, and ldap_user_class tokens.

-s

Creates a services lookaside database file.

-u

Creates a users lookaside database file.

Note: You can specify the -n switch in conjunction with the -u switch to add user data that is collected from an LDAP service.

-G

Lists the contents of the groups lookaside database files.

-H [IPv4 | IPv6]

Lists the contents of the hosts lookaside database files.

-S

Lists the contents of the services lookaside database files.

-U

Lists the contents of the users lookaside database files.

options

Specifies optional modifiers that change the way the utility displays its information. Can be one or more of the following:

-l

Loads the lookaside database using only the list file. This excludes the resolution mechanism of the system.

-f

Fast loads the lookaside database (hosts only) when using the -h switch.

More information:

The seos.ini Initialization File (see page 287)

sechkey Utility

Use the sechkey utility to manage CA Access Control encryption and so protect your CA Access Control management communications. You must have the ADMIN attribute to use sechkey.

You can use it to set an encryption key for symmetric encryption, or you can use it for SSL (PKI) encryption.

If you are using symmetric keys, we recommend that you change the key from the default. If you are using SSL, we recommend that you change the default certificate and associated private key from the default.

Whichever encryption method you use, change the keys on all computers at your site after you have installed or upgraded CA Access Control. This prevents unauthorized users from accessing the system.

The utility handles the following tasks:

- Change a symmetric encryption key (see page 109)
- Change the symmetric encryption method (see page 110)
- Configure X.509 certificates (see page 112)
- Change the Message Queue password (see page 115)

sechkey Utility—Change a Symmetric Encryption Key

The sechkey utility changes the CA Access Control symmetric encryption key for CA Access Control programs.

You can run sechkey in interactive or non-interactive mode. When you run sechkey in interactive mode, sechkey prompts you to enter the old and new encryption keys.

You must stop CA Access Control before you use sechkey to change a symmetric encryption key. You must have the ADMIN attribute to use sechkey.

Important! To avoid communication problems, use the same encryption key on all computers that run CA Access Control components.

This utility has the following format in interactive mode:

```
sechkey
```

This utility has the following format in non-interactive mode:

```
sechkey {oldkey | -d} {newkey | -d} [-s registry_path]
```

sechkey has some additional switches that are only valid on UNIX computers. This utility has the following format for UNIX computers:

```
sechkey {oldkey | -d} {newkey | -d | -n} [-nopmd | -r hostname]

sechkey -k newkey

sechkey -c
```

-c

(UNIX) Clears the selogrd encryption key. The default key is saved in the key file.

Note: The saved key itself is encrypted with the default encryption method.

-d

Specifies the default CA Access Control key.

-k

(UNIX) Specifies the selogrd encryption key that you want to change to. The encryption key is saved in a new file or updated in the old one.

-n

(UNIX) Lists the programs that are using the current key, without changing to a different key.

newkey

Specifies the new encryption key.

-nopmd

(UNIX) Changes the key without updating the Policy Model update file with the new key.

oldkey

Specifies the (current) encryption key that you want to change.

-r *hostname*

(UNIX) Specifies the name of the remote computer whose encryption key you want to change.

To use this option, CA Access Control must be running on both the local and remote computers. This parameter does not actually change the key; rather, it saves information so that the next time you start CA Access Control on the remote computer (using seload -c), the key is changed.

-s *registry_path*

(Windows) Specifies the registry root path where the encryption key for CA Access Control programs is stored. This switch is only valid for third-party programs that use the CA Access Control SDK.

Example: Check If a UNIX Computer Uses the Default Encryption Key

The following command checks if a UNIX computer uses the default CA Access Control encryption key:

```
sechkey -d -n
```

sechkey Utility—Change the Symmetric Encryption Method

The sechkey utility changes the symmetric encryption method for CA Access Control programs. When you change the symmetric encryption method, sechkey decrypts each encrypted password in the CA Access Control database then encrypts each password with the new encryption method.

Note: If CA Access Control is operating in FIPS-only mode, you cannot change the symmetric encryption method. CA Access Control operates in FIPS-only mode when the value of the fips_only configuration token in the crypto section is 1. This restriction prevents you from changing the encryption method to a non-FIPS compliant method.

You must stop CA Access Control before you use sechkey to change the symmetric encryption method. You must have the ADMIN attribute to use sechkey.

Important! To avoid communication problems, use the same encryption method on all computers that run CA Access Control components.

This utility has the following format:

```
sechkey -m -sym {aes128 | aes192 | aes256 | des | tripledes | default} [-s registry_path]
```

-m

Specifies to change the encryption method.

-s *registry_path*

(Windows) Specifies the registry root path where the encryption key for CA Access Control programs is stored. This switch is only valid for third-party programs that use the CA Access Control SDK.

-sym

Specifies the new encryption method to use.

aes128

Specifies to use the following encryption method:

(Windows): aes128enc.dll

(UNIX): libaes128.so

aes192

Specifies to use the following encryption method:

(Windows): aes192enc.dll

(UNIX): libaes192.so

aes256

Specifies to use the following encryption method:

(Windows): aes256enc.dll

(UNIX): libaes256.so

des

Specifies to use the following encryption method:

(Windows): desenc.dll

(UNIX): libdes.so

tripledes

Specifies to use the following encryption method:

(Windows): tripledesenc.dll

(UNIX): libtripledes.so

default

Specifies to use the following proprietary CA Access Control encryption method:

(Windows): defenc.dll

(UNIX): libscramble.so

Example: Change the Symmetric Encryption Method to AES256

The following command changes the symmetric encryption method to AES256:

```
sechkey -m -sym aes256
```

More information:

ChangeEncryptionMethod Utility—Change Encryption Method (see page 27)

sechkey Utility—Configure X.509 Certificates

The sechkey utility configures the root and server certificates that CA Access Control uses to authenticate communication between components.

You can use the sechkey utility to perform the following tasks:

- Configure CA Access Control to use third-party root and server certificates, including OU password-protected certificates
- Create a server certificate from a third-party root certificate
- Save the password of a password-protected certificate on the computer

You must stop CA Access Control before you use sechkey to configure X.509 certificates. You must have the ADMIN attribute to use sechkey.

Note: If CA Access Control is operating in FIPS-only mode, you cannot use password-protected certificates. CA Access Control operates in FIPS-only mode when the value of the fips_only configuration token in the crypto section is 1. This restriction prevents you from encrypting passwords within the certificate with a non-FIPS compliant method.

This command has the following format to create an X.509 root or server certificate:

```
sechkey -e {-ca|-sub [-priv privfilepath]} [-in infilepath] [-out outfilepath]
[-capwd password] [-subpwd password]
```

This command has the following format to use OU password-protected server certificates:

```
sechkey -g {-subpwd password | -verify}
```

-ca

Specifies that sechkey creates a self-signed certificate that is used as a CA (root) certificate.

sechkey stores the certificate and private key in the PEM file defined by the ca_certificate configuration setting in the crypto section.

-capwd *password*

Specifies the password for the private key of the root certificate that sechkey uses to generate a server (subject) certificate.

-e

Specifies that sechkey creates an X.509 certificate.

-g

Specifies that CA Access Control uses third-party server certificates. Save the third-party server certificate in the location specified in the subject_certificate configuration setting in the crypto section, or edit the value of the subject_certificate configuration setting in the crypto section to specify the full path to the third-party server certificate.

Note: If you install the server certificate in a new directory, write CA Access Control FILE rules to protect the new directory.

-in *infilepath*

Specifies the input file that contains the certificate information. If -in is not specified, sechkey reads the information from the standard input.

sechkey requires the following information to create a certificate:

- Serial Number
- Subject
- Not Before (First valid day for certificate)
- Not After (Last valid day for certificate)

sechkey can use the following information, but the information is not mandatory:

- Email
- URI (often named URL)
- DNS name
- IP Address

-out *outfilepath*

Specifies the output file to put the certificate information. The output file is a copy of the input information. If -out is not specified, sechkey does not duplicate the input information.

-priv *privfilepath*

Specifies the file that holds the private key associated with the certificate. This option is only valid when used with the -sub option.

-sub

Specifies that sechkey creates a server (subject) certificate.

sechkey stores the certificate and private key in the PEM file defined by the subject_certificate configuration setting in the crypto section.

If -priv is not specified, the private_key configuration setting in the crypto section defines the file that holds the private key associated with the certificate.

If you create a password-protected server certificate, sechkey does not encrypt the certificate. If you create a server certificate that is not password-protected, sechkey encrypts the certificate using AES256 and the CA Access Control encryption key.

-subpwd *password*

Specifies the password for the private key of the server (subject) certificate. sechkey stores the password in the crypto.dat file in the *ACInstallDir*/Data/crypto directory, where *ACInstallDir* is the directory in which you installed CA Access Control. The crypto.dat file is hidden, encrypted, read-only, and protected by CA Access Control. If CA Access Control is stopped, only the superuser can access the password.

-verify

Verifies that CA Access Control can use the stored password to open the password-protected server key.

Example: Create a Server Certificate from an OU Password-Protected Third-Party Root Certificate

The following command creates a server certificate from an OU password-protected third-party root certificate, using the following values:

- The path to the input file that contains the certificate information is C:\Program Files\CA\AccessControl\data\crypto\sub_cert_info
- The path to the private key for the root certificate is C:\Program Files\CA\AccessControl\data\crypto\ca.key
- The password for the private key for the root certificate is P@ssw0rd

```
sechkey -e -sub -in "C:\Program Files\CA\AccessControl\data\crypto\sub_cert_info"
-priv "C:\Program Files\CA\AccessControl\data\crypto\ca.key" -capwd P@ssw0rd
```

Example: Input File

The following is an example of an input file that contains certificate information:

```
SERIAL: 00-15-58-C3-5E-4B
SUBJECT: CN=192.168.0.1
NOTBEFORE: “12/31/08”
NOTAFTER: "12/31/09"
E-MAIL: john.smith@example.com
URI: http://www.example.com
DNS: 168.192.0.100
IP:  168.192.0.1
```

sechkey Utility—Change the Message Queue Password

The sechkey utility lets you change the Message Queue password. You can change the client or server Message Queue password.

You must have the ADMIN attribute to use sechkey.

This command has the following format:

```
sechkey -t [-server] -pwd password
```

-t

Specifies to change the Message Queue password.

-server

Specifies to change the server Message Queue password.

Note: If you do not specify this parameter, sechkey changes the client Message Queue password.

-pwd *password*

Defines the new password.

More information:

acuxchkey Utility—Change Encryption Key Settings (see page 26)

seclassadm Utility—Administer CA Access Control Classes

The seclassadm utility manages CA Access Control classes. It adds new user-defined classes to the local database. Invoke it from the directory in which the database resides (or use the -p option), while CA Access Control is *not* running.

Note: Running seclassadm creates a file in the seosdb directory with the new class information. When you create a new database with dbmgr -c, user-defined classes are created in the new database if the CreateNewClasses configuration setting is set to yes (the default).

This command has the following format:

```
seclassadm -add className [-a access] [{-|+}c] [-d access] \
    [-f] [-g] [-o] [-p db_pathname] [-t]

seclassadm -del className

seclassadm -upd className {-|+}c [-p db_pathname]
```

-add *class-name*

Adds a new resource class to an existing database, where *class-name* is the name of the new class.

CA Access Control reserves class names that are in uppercase characters. When adding a class, use at least one lowercase character. Class names can be up to 79 characters long.

After creating a new class, you must enable the class by using the selang setoptions command.

-del *class-name*

Deletes the specified resource class from the database.

-upd *class-name*

Updates the specified resource class in the database.

-a *access*

Specifies the access modes for the class. The string *access* represents the allowed accesses. Each access mode is represented by a single character code listed in any order. The string must not contain any blank or other non-alphabetic characters. Valid access modes are:

Abbreviation	Description
C	control
D	delete
E	create
F	filescan
M	chmod
O	chown
R	read
S	security
T	utime
U	update
V	rename
W	write
X	execute

-d *access*

Specifies the default access mode for the class. This is the access mode that CA Access Control assigns to a user when you execute the authorize command without specifying an access authority. This implicit access used by the authorize command is *not* the same as the default access assigned to a resource. The possible accesses modes are listed in the -a option.

-f

Specifies that CA Access Control will accept a new class name, even if the name contains all upper case letters.

Note: By default, the seclassadm utility does not let you create a class name that is all uppercase. CA Access Control uppercase names are reserved for the predefined CA Access Control classes.

-g

Specifies that the new class is a resource that groups members of an existing class. The relationship between the existing class and the new group class is the same as the relationship between any class and its group class in the database (for example, TERMINAL and GTERMINAL). A resource that groups members of an existing class must begin with the upper case letter G. That is, it has the same name as the existing class, but begins with the prefix *G*.

-o

Creates a *_default* record for the new class and sets its default access.

-p *db_pathname*

Specifies the full pathname of the local database.

By default, the utility works on the database in the current directory. Use this option to define a different directory where the database resides.

-t

Specifies that this is a Unicenter TNG class.

Examples: Add a new class to the database

The following examples demonstrate how you can use the seclassadm utility to add a class to the database:

- To add a resource class by the name *dbfield*, use the following command:

  ```
  seclassadm -add dbfield
  ```

- To add a resource class by the name *report* with only READ access, use the following command:

  ```
  seclassadm -add report -d R -a R
  ```

- To add a resource class by the name *batch_jobs* with READ, WRITE, and MODIFY permissions and READ access as the default when not specified, use the following command:

  ```
  seclassadm -add batch_jobs -d R -a RWM
  ```

- To add a new class whose objects are groups of resources in the class DEPTA, with access execute and implicit access execute, use the following command:

  ```
  seclassadm -add DEPTA -d X -a X -g -f
  ```

secompas Utility—Compare Passwords

Valid on UNIX

The secompas utility compares passwords in the CA Access Control database with the passwords in the UNIX password file.

For each user in the CA Access Control database, the utility outputs one line that contains the user name and a message indicating whether the user is not defined in UNIX, whether the user has no password in CA Access Control, or whether the passwords match. The utility also displays the total number of users it compared and the number of users whose passwords do not match. It only adds to this total when the password exists in both environments and it is not the same. If a user is not defined in an environment, or the password is missing from an environment, secompas does not add to the counter of unmatched passwords.

To compare passwords, the secompas utility uses the /etc/passwd file, the shadow password files, and NIS/NIS+ password maps.

Note: You must have the ADMIN attribute to use this utility.

This command has the following format:

```
secompas [-db] [-ok] [-ux]
```

-db

Specifies not to display users that do not have a password in the CA Access Control database.

-h

Displays the help for this utility.

-ok

Specifies not to display users that have the same password in the CA Access Control database and UNIX (password match).

-ux

Specifies not to display users that do not exist in UNIX.

Example: Utility output

This example shows sample output from the utility:

```
Checking root           : No password in Access Control database.
Checking tst_001        : Undefined in UNIX.
Checking tst_002        : No password in UNIX password file
Checking tst_003        : *** PASSWORDS DO NOT MATCH. ***
Checking tst_004        : *** NO MATCH - UNIX DISABLED ***
Checking tst_005        : OK

Total of 6 users found in database.
2 unmatched password(s) found. (1 UNIX DISABLED).
```

The following explains each line in the preceding output:

```
Checking root           : No password in Access Control database.
```

Either the user *root* is not defined in the CA Access Control database or the user is defined in the database but does not have a password in it.

```
Checking tst_001        : Undefined in UNIX.
```

The user tst_001 is defined in the CA Access Control database but not in UNIX.

```
Checking tst_002        : No password in UNIX password file
```

The user tst_002 is defined in UNIX but does not have a password.

```
Checking tst_003        : *** PASSWORDS DO NOT MATCH. ***
```

The CA Access Control password does not match the UNIX password of the user tst_003.

```
Checking tst_004        : *** NO MATCH - UNIX DISABLED ***
```

The tst_004 user account was disabled in the UNIX environment. secompas identifies a disabled user account by the asterisk (*) in front of the password in the /etc/passwd file.

```
Checking tst_005        : OK
```

The CA Access Control password matches the UNIX password of the user tst_005.

secons Utility

The secons utility is the CA Access Control security console. It lets you perform the following tasks:

- On UNIX:
  - Display run-time statistics (see page 135)
  - Manage concurrent login options (see page 126)
  - Manage CA Access Control tracing (see page 125)
  - Manage resource caching (see page 127)
  - Manage CA Access Control shutdown (see page 122)
  - Reload configuration settings (see page 150)
  - Remove XUSER objects (see page 132)
  - Display kernel tables (see page 140)
  - Clean, enable, or disable kernel cache tables (see page 149)
- On Windows:
  - Control instrumentation run-time settings (see page 151)
  - Display run-time statistics (see page 138)
  - Display ACEE Records (see page 133)
  - Manage concurrent login options (see page 126)
  - Manage CA Access Control tracing (see page 125)
  - Refresh IP addresses for network resources (see page 150)
  - Remove XUSER objects (see page 132)
  - Resolve recycled accounts (see page 134)
  - Shut down CA Access Control (see page 132)
  - Display your user name and security credentials (see page 154)

The secons utility is available to both security administrators and other users. However, only some options are available for users who do not have the ADMIN attribute. These options are:

-m (trace management), -d-, -d+, -ds (login management), and -whoami (user's credentials).

secons Utility—Manage CA Access Control Shutdown on UNIX

Valid on UNIX

The secons utility shuts down CA Access Control and the associated daemons. You can also use this utility to find out which processes are still executing CA Access Control code.

Only users defined as ADMIN or OPERATOR can shut down CA Access Control. To shut down CA Access Control on remote computers, you must be defined as ADMIN or OPERATOR on those remote computers.

This command has the following format:

```
secons [-s [hosts | ghosts]] \
    [-S [{selogrd | selogrcd | serevu}]] \
    [-sc] [-scl] [-sk]
```

-s [*hosts* | *ghosts*]

Shuts down the CA Access Control daemons on the defined, space-separated, list of remote hosts. If you do not specify any hosts, CA Access Control shuts down on the local host.

You can define a group of hosts by entering the name of a *ghost* record. If you use this option from a remote terminal, the utility requests password verification. You also need admin privileges on both the remote and local computers, and write permission to the local computer on the remote host database.

-S [{selogrd | selogrcd | serevu}]

If you do not define a daemon, terminates the CA Access Control daemons and attempts to terminate active daemons selogrd, selogrcd and serevu. If the selogrd, selogrcd, or serevu tokens in the [daemons] section of seos.ini file are set to *yes*, sends the termination request to the running CA Access Control main daemon or sends the termination signal to the specified daemon if CA Access Control is already down.

If you define a daemon, secons does not terminate the CA Access Control daemons. If the appropriate token in the [daemons] section of seos.ini file is set to *yes*, it sends the termination request to the running CA Access Control main daemon or it sends the termination signal to that daemon if CA Access Control is down.

-sc[l]

Displays processes that are still executing CA Access Control code.

You cannot unload CA Access Control if an application, which is loaded on top of CA Access Control, has an open system call (syscall) that is hooked by CA Access Control. Once you know which processes are still executing CA Access Control code, you can shut down these processes and unload the CA Access Control kernel module. You can then use UNIX exits to automatically shut down these processes before unloading the kernel and then restart them after the kernel unloaded.

The *-sc* output displays as a two-column table with the system call number in the first column, and the process identifier in the second column.

The *-scl* option also displays parent process ID (PPID), UID, time, and program name information for the processes that are still executing CA Access Control code. The time information lets you find out how long the process has CA Access Control hooked. If the time is relatively short, the hook is likely to be a temporary one.

You can also run this while CA Access Control is running to help you predict what may cause unload issues in advance. However, in some cases, such as the accept command, CA Access Control code removes the hook during unload. This means that some of the active hooks you see while CA Access Control is running may not actually affect unloading.

Note: By default, CA Access Control monitors system calls intercepted by CA Access Control. You must set the syscall_monitor token in the seos.ini file to 0 (disabled) if you do *not* want CA Access Control to monitor system calls.

-sk

Shuts down all CA Access Control daemons and prepares the CA Access Control kernel extension to be unloaded.

Example: Shut Down CA Access Control

- To shut down the CA Access Control daemon, enter:

  ```
  secons -s
  ```

- To shut down the CA Access Control daemon on remote hosts HOST1 and HOST2, enter:

  ```
  secons -s HOST1 HOST2
  ```

Example: Display Information for Processes that are Still Executing CA Access Control Code

- To display basic information on processes that are still executing CA Access Control code:

  ```
  secons -sc
  ```

  The output you receive looks similar to the following:

  ```
  CA Access Control secons vX.X.X.xxx - Console utility
  Copyright (c) YYYY CA. All rights reserved.
  Active system calls:

  syscall    5 - PID:  27477
  ```

- To display more information on processes that are still executing CA Access Control code:

  ```
  secons -scl
  ```

  The output you receive looks similar to the following:

  ```
  CA Access Control secons vX.X.X.xxx - Console utility
  Copyright (c) YYYY CA. All rights reserved.
  Active system calls:

  -Syscall  102 - PID:   2105  PPID:     1 UID:     0 TIME:    4d-4h PROGRAM NAME:
  /usr/sbin/vsftpd
   Syscall    5 - PID:  24269  PPID:   4289 UID:     0 TIME:   2d-21h PROGRAM NAME:
  /bin/bash
  ```

  A dash (-) at the beginning of the output line means that CA Access Control assesses that this hook is not likely to cause you issues when unloading. When you use this command, CA Access Control also adds lines to the audit log that records whether the unloading CA Access Control is likely to succeed. For example, the following audit record is created when you run secons -scl and there is at least one blocking system call that is likely to prevent CA Access Control from unloading:

  ```
  10 Nov 2008 05:47:22 F CHECK      root      Scan     339  0 SEOS_syscall    unload
  ```

secons Utility—Manage CA Access Control Tracing

The secons utility manages CA Access Control tracing. Tracing lets you monitor operating system events. CA Access Control can accumulate a file of messages reporting operating system events that you can then display.

This command has the following format:

```
secons [-t+] [-t-] [-tt] [-ts] [-tc] [-tv [size] [-file fileName]]

secons -m message

secons -pupm trace {enable | disable | clear}
```

-m *message*

Adds a text message to the trace file.

-t+

Enables tracing, which causes the CA Access Control engine (seosd) to dump messages that specify its operations and actions to the trace file.

-t-

Disables tracing, which stops the CA Access Control engine seosd from dumping messages to the trace file.

-tc

Clears the trace file, removing all records from the it.

Note: You can use this option whether or not seosd is running.

-ts

Displays the current tracing status.

-tt

Toggles the tracing status.

-tv [*size*] [-file *fileName*]

Displays a real-time trace output. The utility displays the last *size* KB (by default, 2 KB) of the trace file and keeps the session open so that any new trace messages added to the file are displayed. This is similar to the UNIX tail -f command.

Use Ctrl+C to stop this operation.

Note: You can use this option whether or not seosd is running. Use the full_year configuration setting to select whether you want to display the year in four digits (the default, yes) or two digits.

size

Specifies the size, in kilobytes, of the portion of the file you want to display, starting from the end. Specify 0 to show the entire trace file. If you do not specify this option, secons uses the default, 2 KB.

-file *fileName*

Reads *fileName* instead of *ACInstallDir*/log/seosd.trace.

-pupm trace {enable | disable | clear}

Valid for the Privileged User Password Management Agent

Specifies tracing options on the Privileged User Password Management Agent during runtime. You do not need to restart CA Access Control to modify the trace options.

Limits: enable, enables tracing; disable, disables tracing; clear, clears the trace file.

Important! The trace options you specify apply to the current session only. After CA Access Control restarts the trace option is set according to the OperationMode token in the PUPMAgent section.

secons Utility—Manage Concurrent Login Options

The secons utility manages concurrent login options. You can configure CA Access Control to prevent a user from logging in more than once. This prevents intruders from logging into the accounts of users who are already logged in.

This command has the following format:

```
secons [-d+] [-d-] [-ds] [-l+] [-l-] [-ls] \
        [-u+ userName] [-u- userName] [-us userName]
```

-d+

Enables concurrent logins for the user running the command.

-d-

Disables concurrent logins for the user running the command. Using this command kills any concurrent logins of the user to the local computer.

Note: You can also place this command in the .login or .cshrc file of a user to disable concurrent logins.

-ds

Displays the concurrent logins setting for the user running the command.

-l+

Enables concurrent logins system-wide.

Note: By default, CA Access Control enables login, but in cases where the system is shut down for maintenance, you can disable login for a specific period.

-l-

Disables concurrent logins system-wide.

-ls

Displays system-wide login status.

-u+ ***userName***

Enables concurrent logins for the defined user.

-u- ***userName***

Disables concurrent logins for the defined user.

-us ***userName***

Displays the concurrent logins setting for the defined user.

secons Utility—Manage Resource Caching on UNIX

Valid on UNIX

The secons utility manages resource caching (file cache) on UNIX. The cache, a runtime table, "remembers" the previous answer to an authorization request (permit or deny) for resources in the FILE class. When an identical authorization is requested, the request is answered with the last response that was stored in the cache memory tables.

This command has the following format:

```
secons [-C+] [-C-] [-CA value] [-CC interval] [-CD] \
    [-CF value] [-CI init_value] [-CP interval] -CU value]
```

-C+

Enables caching of file authorization.

-C-

Disables caching of file authorization.

-CA *value*

Specifies the maximum number of authorization records in a table.

Default: 80

Limits: A number between 1 and 800

-CC *interval*

Specifies the cache clean interval in minutes.

Default: 60

Limits: A number greater than 0

-CD

Displays the cache table to the standard output.

-CF *value*

Specifies the maximum number of file records in a table.

Default: 20

Limits: A number between 1 and 200

-CI *init_value*

Specifies the initial priority value for a new record in the cache table.

Default: 10

-CP *interval*

Specifies the cache priority computing interval.

Default: 1 (one record)

Limits: A number between 1 and 10

-CU *value*

Specifies the maximum number of user records in a table.

Default: 50

Limits: A number between 1 and 500

Example: Change cache settings

The following example shows you how you can change settings of the cache so that the maximum number of file, user, and authorization records in the cache are 60:

```
secons -CF 60 -CU 60 -CA 60
```

Example: Display the cache table

The following example shows the output of the secons -CD command:

```
================================================================================
        FILE CACHE (configuration, statistics, and dispatcher data)
--------------------------------------------------------------------------------
sizes(bytes)       tables:               | max records:      | intervals
cache   head      files    users    auth  | files users auths |clean prio
--------------------------------------------------------------------------------
40244    44        5600     4200    30400 |  20    50    80   |   60   1
================================================================================
table  |statistics        | priority    |min | rec  | average      |pri |init
name   |  hits misses (ok)| maxim  minim|ind | used | usage   life |fact|prio
--------------------------------------------------------------------------------
files  |    5     1   83% |   0      0  |  0 |   1  |              |    |
users  |    5     1   83% |  10      2  |  0 |   1  |  0       0   |  1 | 10
auths  |    4     2   66% |   2         |  0 |   2  |              |    |
================================================================================
FILE TABLE
--------------------------------------------------------------------------------
No  type    pid priority user                                 file name
--------------------------------------------------------------------------------
0   EXPL    372      0    0                                      /etc/shadow
================================================================================
USER TABLE
--------------------------------------------------------------------------------
No   user name     prio   life   used   UID  EUID  RUID auth prev(file)next
--------------------------------------------------------------------------------
0    root           2       2      7     0    0     0    0    50(   0)  50
================================================================================
AUTHORIZATION RESULT TABLE  (R - Result: 'P'-permit, 'D'-deny ...)
--------------------------------------------------------------------------------
No  R ACEE acc  Log stage prv(usr)nxt time       terminal  program
--------------------------------------------------------------------------------
0   P   6  read  0  00036 80(  0)  1   07:48:25            /usr/bin/login
================================================================================
```

The following explains the preceding output:

The output consists of five parts:

- The cache configuration. It contains the following fields:
  - Size of the cache (in bytes)
  - Size of the cache header (in bytes)
  - Size of the file table (in bytes)
  - Size of the user table (in bytes)
  - Size of the results table (in bytes)
  - The maximum number of file records
  - The maximum number of user records
  - The maximum number of result records
  - Statistic: hits in the table
- The table of file records. It contains the following fields:
  - Sequential number of the record
  - Type of the file (EXPLICIT, IMPLICIT)
  - Process ID number
  - Priority of the record, is sum of its users priorities
  - Appropriate user record number in the table of users
  - Name of the file
- The table of users. It contains the following fields:
  - Sequential number of the record
  - User name
  - Priority of the record
  - Record lifetime counter
  - Record usage counter
  - User ID; user effective ID; really used by security ID
  - Appropriate authorization record number in the table of authorization
  - Previous user record number in the chain of users
  - Appropriate file record number
  - Next user record in the chain of users

- The table of authorization results. It contains the following fields:
  - Terminal
  - Stage
  - Granted stage
  - Result - authorization result (P or D)
  - ACEE number
  - Access type
  - Logging options flag value
  - The stage number the decision was made
  - Previous authorization record number in the chain of records
- Appropriate user record number
  - Next authorization record number in the chain of records
  - Statistic: the number of missed records in the table
  - Authorization class
  - Program name (with the via parameter)
  - Notification string
  - Update time (GMT)
- Dispatcher Data. It contains the following fields:
  - Statistic: number of missed records in the table
  - Statistic: number of hits in the table
  - Maximum priority in a table
  - Minimum priority in a table
  - Number of entries with minimum priority
  - Number of used records
  - Average usage (only for users table)
  - Average life (only for users table)
  - Priority calculation factor (only for users table)
  - Initial value of the record priority (only for users table)

secons Utility—Shut Down CA Access Control on Windows

Valid on Windows

The secons utility shuts down the CA Access Control engine and all other CA Access Control services on the local station or on one or more remote stations.

Only users defined as ADMIN or OPERATOR can shut down CA Access Control. To shut down CA Access Control on remote computers, you must be defined as ADMIN or OPERATOR on those remote computers.

This command has the following format:

```
secons -s [hosts | ghosts]
```

-s [*hosts* | *ghosts*]

Shuts down the CA Access Control services on the defined, space-separated, remote hosts. If you do not specify any hosts, CA Access Control shuts down on the local host.

You can define a group of hosts by entering the name of a ghost record. If you use this option from a remote terminal, the utility requests password verification. You also need admin privileges on both the remote and local computers, and write permission to the local computer on the remote host database.

secons -dbclean—Remove XUSER Objects from the CA Access Control Database

The secons utility removes XUSER objects that were not resolved to their native security identifiers (SID) from the CA Access Control database. Use the secons -dbclean command to remove XUSER objects that no longer exist in the native environment.

This command has the following format:

```
secons –dbclean <osuser>
```

-dbclean

Specify to remove all XUSER objects that were not resolved from the CA Access Control database.

<osuser>

Specifies the native user account name.

secons -acee Function—Display ACEE Records on Windows

Valid on Windows

The secons utility lets you monitor the Accessor Element Entry (ACEE) table that caches accessors in the authorization engine. The ACEE stores information about the following users:

- **Logged in user**—A user that has logged in to the operating system. Specific ACEE attributes for this type of user are:
  - Login session ID
  - Login session type
- **Management user**—A user that has logged in to a CA Access Control management application (using an LCA connection). For example, selang.
- **Authorization API user**—A user that was referenced in SEOSROUTE_* API.
- **SPECIALPGM Logical user**—A user that is being references at least in one SPECIALPGM record. A specific ACEE attribute for this type of user is:
  - ACEE association with SPECIALPGM records
- **Built in user**—A user that is built in CA Access Control. For example, *_undefined*.

Note: Only a CA Access Control administrator can use this command.

This command has the following format:

```
secons -acee [handle | all | list]
```

all

Displays all ACEE records.

handle

Defines the ACEE handle you want to display.

list

Displays a summary list of all ACEE records, without the full details.

Examples: Display ACEE Records

- This example displays a list of handles in the ACEE:

  ```
  secons -acee list
  ```

  The secons output looks like this:

  ```
  ACEE handle '0' represents 'Logged on User': NT AUTHORITY\ANONYMOUS LOGON (OS
  User)
  ACEE handle '1' represents 'Logged on User': NT AUTHORITY\NETWORK SERVICE (OS
  User)
  ACEE handle '2' represents 'Logged on User': COMP1-SRV-X86\John
  ACEE handle '3' represents 'Logged on User': NT AUTHORITY\LOCAL SERVICE (OS User)
  ACEE handle '4' represents 'Logged on User': NT AUTHORITY\SYSTEM (OS User)
  ACEE handle '5' represents 'Management User': COMP1-SRV-X86\John
  ACEE handle '6' represents 'SPECIALPGM Logical User': logicaluser
  ```

- This example displays handle 6 in the ACEE:

  ```
  secons -acee 6
  ```

  The secons output looks like this:

  ```
  ACEE handle '6' represents 'SPECIALPGM Logical User': logicaluser
  ACEE was created at: Wed Feb 20 17:35:52 2008
  ACEE was last accessed at: Wed Feb 20 17:35:52 2008
  ACEE user role is: Regular
  ACEE audit mode is: Failure, Login Success, Login Failure; Originated from User
  definition
  ACEE user is a member of 0 'CA Access Control' groups
  ACEE user is associated with 1 SPECIALPGM records
      1. C:\WINDOWS\system32\calc.exe
  ```

secons -checkSID Function—Resolve Recycled Accounts on Windows

Valid on Windows

The secons utility compares the security identifier (SID) of each enterprise account (XUSER and XGROUP resource) with the native Windows account SID, and creates a backup of recycled accounts. As the CA Access Control authorization is based on SID, where the SID of a CA Access Control accessor resource differs from the native account SID (a recycled account), the utility creates a new account (with the same name as the old account) and backs up the obsolete recourse using the following naming convention: *SID* (*accountName*)

Note: For more information on recycled enterprise store accounts, see the *Endpoint Administration Guide for Windows*.

This command has the following format:

```
secons -checkSID {-groups | -users} [accountName [,accountName...]]
```

-groups

Specifies that secons should examine enterprise group records.

-users

Specifies that secons should examine enterprise user records.

accountName

Specifies the name of a user or group that secons should search for. If *accountName* is omitted, secons looks for all groups or users.

secons -i Function—Display Run-time Statistics on UNIX

Valid on UNIX

The secons utility displays CA Access Control run-time statistics about system behavior. Use this information to learn about network connection requests, the size of the audit and error log queues, size of cached tables, and the size of the database and number of records in each part of the database.

This command has the following format:

```
secons -i
```

-i

Displays runtime statistics as formatted text.

Example: Display run-time data

The following example shows the output of the secons -i command:

```
Runtime Statistics:
          --------------------
          INet statistics:
                Requests denied   : 0
                Requests granted  : 17
                Errors found      : 0
          Queues size:
                Audit log: 0
                Error log: 0
          Cached tables info:
                ACEE handles      :      11
                Protected clients :      0
                Trusted programs  :      77
                Untrusted programs:      3
          Database info: (Record count & first free ID)
                Classes     :    235 ( CID     0x00f0 )
                Properties  :   4829 ( PID     0x1346 )
                Objects     :    842 ( OID 0x0000035a )
                PropVals    :   4109 ( N/A )

CA Access Control memory utilization statistics:
-----------------------------------------------
CA Access Control security daemon (seosd, pid=4265 Total=13MB Res=8MB)
CA Access Control agent daemon (seagent, pid=4268 Total=10MB Res=3MB)
CA Access Control watchdog daemon (seoswd, pid=4276 Total=6MB Res=3MB)
CA Access Control policyfetcher daemon (policyfetcher, pid=4331 Total=11MB Res=1MB)
CA Access Control AgentManager daemon (AgentManager, pid=4561 Total=22MB Res=3MB)
```

The following explains each line in the preceding output:

```
INet statistics:
      Requests denied   : 0
      Requests granted  : 17
      Errors found      : 0
```

Displays statistics of network access authorization performed by CA Access Control. These lines summarize the number of denials, grants, and errors during the authorization of network requests.

```
Queues size:
      Audit log: 0
      Error log: 0
```

Since CA Access Control creates logging with file locking, it is possible that certain events are held in memory and written to log files after a while. If these values exceed 10, then an error could be interfering with the CA Access Control logging facility.

```
Cached tables info:
      ACEE handles      :      11
      Protected clients :      0
      Trusted programs  :      77
      Untrusted programs:      3
```

Displays information about the size of cached tables CA Access Control uses:

- Accessor Element Entry (*ACEE)* is a table containing logged-in processes.
- *Protected clients* lists the number of cached clients. Usually, this value is 0.
- *Trusted Programs* lists the number of entries in class PROGRAM that are cached in memory. Normally, all programs should be cached as trusted.
- *Untrusted Programs* displays the number of programs that were found to be untrusted.

```
Database info: (Record count & first free ID)
      Classes     :    235 ( CID     0x00f0 )
      Properties  :   4829 ( PID     0x1346 )
      Objects     :    842 ( OID 0x0000035a )
      PropVals    :   4109 ( N/A )
```

General information regarding the size of the database and the number of records in each part of the database.

Displays information about the size of the memory the following daemons use: seosd, seagent, seoswd, policyfetcher and AgentManager:

```
CA Access Control memory utilization statistics:
------------------------------------------------
CA Access Control security daemon (seosd, pid=4265 Total=13MB Res=8MB)
CA Access Control agent daemon (seagent, pid=4268 Total=10MB Res=3MB)
CA Access Control watchdog daemon (seoswd, pid=4276 Total=6MB Res=3MB)
CA Access Control policyfetcher daemon (policyfetcher, pid=4331 Total=11MB Res=1MB)
CA Access Control AgentManager daemon (AgentManager, pid=4561 Total=22MB Res=3MB)
```

The CA Access Control watchdog daemon monitors the memory use of the daemons based on the interval defined in the ProcVSizeInterval token. If a process exceeds the specified watermark in the ProcVSizeHigh token, the watchdog restarts the process within the time frame defined in the ProcRestartHours token. If a process exceeds the memory size specified in the ProcVSizeCritical token, the watchdog daemon immediately restarts the process.

More information:

seoswd (see page 362)

secons -i Function—Display Run-time Statistics on Windows

Valid on Windows

The secons utility displays CA Access Control run-time statistics and internal counters. Use this statistical system behavior information to learn the following:

- How many events were triggered for each interception type.
- How effective each kernel cache is , by comparing the number of cached events against the number of fully authorized events.

Note: It is normal for the audit queue to increase in periods of increased activity. However, the queue size should decrease once the load is normal again.

This command has the following format:

```
secons -i [-reset]
```

-i

Displays runtime statistics as formatted text.

-reset

(Optional) Resets the run-time counters to zero.

Example: Display run-time data

The following describes the information that is not self-explanatory in the output of the secons -i command:

Database run-time data

Displays the number of classes, objects, and properties in the CA Access Control database, the ID of the last created class, object, and property, and the number of property values.

Use this information to evaluate the size of the database. The more objects and properties you use, the bigger the database is.

Kernel run-time data

Displays for each of the kernel caches (file, registry, and surrogate) their creation time, size, and efficiency. Efficiency is the number of audit events out of the total number of events. The remaining interception events follow the authorization process.

Use this information to evaluate the need for, and efficiency of, each kernel cache.

Kernel audit information

Displays the current kernel audit queue size and the maximum size it reached and when.

Use this information to evaluate the audit queue behavior. You should make sure that the audit queue does not exceed the maximum allocated queue size, which is set in the FsiDrv\MaxAuditRecordLimit CA Access Control registry entry. When this limit is reached, CA Access Control generates audit events more slowly so that the queue can be resolved.

User mode enforcement run-time data

Displays information for intercepted file, registry, logon, kill, and Windows service events in Full Enforcement mode. You can find out about the number of events being authorized by the authorization engine and the maximum and average time an authorization process took to complete for each class.

Use this information to troubleshoot problems in a live production system. It provides you with some valuable initial data without needing to shut down CA Access Control.

User mode audit run-time data

Displays information for audit events (cached intercepted event*)*.

Use this information to monitor user mode audit queue behavior. If the maximum audit queue increases consistently, make sure that CA Access Control can write to the audit log file. CA Access Control may not be able to write to the file if the system has run out of disk space, or it does not have native access permissions to file.

Note: It is normal for the audit queue to increase in periods of increased activity. However, the queue size should decrease once the load is normal again.

secons -kt Function—Display Kernel Tables on UNIX

Valid on UNIX

The secons utility displays the kernel tables.

This command has the following format:

```
secons -kt tableNumber
```

-kt

Displays the specified kernel table.

tableNumber

Specifies the kernel table to display. *tableNumber* must be one of the following values:

1

Specifies to display the SpecPgm kernel table.

2

Specifies to display the TrustPg kernel table.

3

Specifies to display the LoginPg kernel table.

4

Specifies to display the DBfiles kernel table.

5

Specifies to display the FRegExp kernel table.

6

Specifies to display the DCMfile kernel table.

7

Specifies to display the AC pids kernel table.

8

Specifies to display the InoCach kernel table.

Note: Not valid on Linux.

9

Specifies to display the F cache kernel table.

10

Specifies to display the NetwDCM kernel table.

11

Specifies to display the MntDirs kernel table.

12

Specifies to display the F inode kernel table.

13

Specifies to display the STOPbyp kernel table.

Note: You cannot display this kernel table if STOP is not enabled.

14

Specifies to display the STOPexp kernel table.

Note: You cannot display this kernel table if STOP is not enabled.

15

Specifies to display the Family kernel table.

16

Specifies to display the DbgProt kernel table.

17

Specifies to display the TCPport kernel table.

18

Specifies to display the TCPoutp kernel table.

19

Specifies to display the ProcSrv kernel table.

Example: Display the DBfiles Kernel Table

The following example shows you an example of the output when you display the DBfiles kernel table:

```
secons -kt 4
DBfiles
file    ID      i-node  device  program name
1       29      280391  356515  /opt/CA/AccessControl/seosdb/seos_ids.dat
2       3       0       0       /opt/CA/AccessControl/etc/privpgms.init
```

Kernel Tables

Kernel tables list frequently-accessed information to help improve CA Access Control performance. Kernel tables improve performance because CA Access Control does not need to check the database to permit, deny, or resolve events that are listed in the kernel tables.

CA Access Control includes the following types of kernel tables:

- Cache tables—List the results of previous resource access requests, resolved inode numbers, and accepted incoming TCP requests.
- Protected resource tables—List resources for which, when access is requested, CA Access Control always sends an authorization request to the CA Access Control engine.
- Bypass tables—List resources for which, when access is requested, CA Access Control permits access without sending an authorization request to the CA Access Control engine.
- Process table—Lists information about all the processes running in the system.

The following table provides information about each kernel table:

Table Name	Type	Lists	Column Names	Configuration Setting
SpecPgm	Protected resource	All objects in the SPECIALPGM class	flags; user; oid; i-node; device; program	SPECIALPGM class records
TrustPg	Protected resource	All objects in the PROGRAM class	flags; i-node; device; program	PROGRAM class records
LoginPg	Protected resource	All objects in the LOGINAPPL class	flags; i-node; device; program name	LOGINAPPL class records
DBfiles	Protected resource	All objects in the FILE class	file ID; i-node; device; program	FILE class records **Note:** The maximum number of records in this table is defined by max_regular_file_rules in the SEOS_syscall section of the seos.ini file

Table Name	Type	Lists	Column Names	Configuration Setting
FRegExp	Protected resource	Generic file access rules that are defined in the FILE class	fid; expression	Defined by a generic rule in a FILE class record **Note:** The maximum number of records in this table is defined by max_general_file_rules in the SEOS_syscall section of the seos.ini file
DCMfile	Bypass	Do-not-call-me files that you define using GAC	fid; user; type; access	GAC.init file
ACpids	Bypass	Process IDs for the CA Access Control daemons	pid; service; contractID	-
InoCach	Cache	Cached inodes	i-node; device; priority; entry	cache_enabled in the SEOS_syscall section of the seos.ini file
F cache	Cache	Cached file access authorization results	file ID; access; acee; answer; phash; prio	-
NetwDCM	Cache	Cached accepted incoming TCP connections	peer; port; local port; flag; prio	UseNetworkCache in the seosd section of the seos.ini file
MntDirs	Protected resource	Directories that CA Access Control protects from mounting	dir ID; i-node; device; mount point	-
F inode	Protected resource	Inode and device number of objects in the FILE class	file ID; i-node; device; links	-

Table Name	Type	Lists	Column Names	Configuration Setting
STOPbyp	Bypass	Objects in the PROGRAM class for which CA Access Control does not provide STOP protection	i-node; device; program	If STOP is enabled, objects in this table have a SPECIALPGM record with the property pgmtype(STOP)
STOPexp	Bypass	Regular expressions that define objects in the PROGRAM class for which CA Access Control does not provide STOP protection	priority; n-chars; expression	If STOP is enabled, objects in this table are defined by a generic rule in a SPECIALPGM record with the property pgmtype(STOP)
Family	Bypass	CA Access Control daemons	service; pid; contractID	-
DbgProt	Protected resource	CA Access Control binaries that CA Access Control protects from debugging	pid; access; name in proc	-
TCPport	Bypass	Ports for which seos_syscall will not pass events to seosd	TCP port	bypass_TCPIP in the seosd section of the seos.ini file
TCPoutp	Bypass	Ports for which seos_syscall will not pass outgoing connection events to seosd	TCP port	bypass_outgoing_TCPIP in the seosd section of the seos.ini file
ProcServ	Process	Lists information about all the processes running in the system	#n; pid; ppid; acee; flags; uid; euid; zone; arg0; ACuser **Note:** There are many more internal columns in this table that are not displayed by the secons utility	-

Kernel Table Column Names

The following list explains the kernel table column names:

#n

Entry number in the kernel table.

access

Defines the type of access that CA Access Control permits, or the type of access that a user requested. The value is a sum of access types:

1-read
2-write
4-chown
8-chmod
16-rename
32-unlink
64-utimes
128-chattr
256-link
512-chdir
1024-create

acee

Defines the ACEE of the user making the access request.

ACuser

Defines the CA Access Control user name of the user.

answer

Defines the response (permit or deny) that CA Access Control made to the access request. Valid values include:

0–deny

1–permit

arg0

Defines the program name, as defined in argument number 0 when the program executes.

contractID

(Solaris 10 only) Defines the contract process ID.

device

Defines the logical disk that the file resides on.

dir ID

Defines the directory ID.

entry

Defines the string value of the inode.

euid

Defines the effective user ID.

expression

Defines the expression (text pattern used for string matching) that specifies the resources to which the entry applies.

fid *or* file ID

Defines the file ID that CA Access Control uses to identify the file.

flags

Defines the bit mask flag for the entry.

i-node

Defines the inode number.

links

Defines the number of hard links of the file.

local port

Defines the port on the local host that accepts the incoming TCP connection.

mount point

Defines the location in the directory to protect from mounting.

n-chars

Defines the number of characters in the expression.

name in proc

Defines the process name in the /proc file system.

Note: In the /proc file system, each process is represented as a file, and the file name is the process number.

oid

Defines the object ID.

peer

Defines the peer host address.

phash

Defines the hash value of a path string.

pid

Defines the process ID.

port

Defines the port from which the incoming TCP connection originated.

ppid

Defines the parent process ID.

prio *or* priority

Defines the priority of the entry in the kernel table. When the kernel table is full, the entry with the lowest priority is removed when CA Access Control writes a new entry to the kernel table.

program *or* program name

Defines the name of the program.

service

Defines the name of the CA Access Control service (daemon).

TCP port

Defines the TCP port to which the entry applies.

type

Defines the protected file type.

uid *or* user

Defines the user ID.

zone

(Solaris 10 only) Defines the zone ID.

Note: The value of this column is always 0 for a non-Solaris 10 computer.

Cache Tables

There are three types of kernel cache tables:

- **F cache**—The file cache table caches the results of previous authorization requests.

  When an identical authorization request is made, CA Access Control answers the request with the last response that is stored in the file cache table.

  Note: The file cache tables is cleaned every 30 minutes and whenever a record changes in the following classes: CALENDAR, CONTAINER, FILE, GFILE, GROUP, HOLIDAY, PROGRAM, SECLABEL, SECLEVEL, SHIFT, and USER.

- **InoCach**—The inode cache table caches resolved inode numbers.

  When CA Access Control needs to resolve an inode number to a file name, it checks the InoCach table before it checks the file system.

- **NetwDCM**—The network cache table stores accepted, incoming TCP requests.

  When CA Access Control receives an incoming TCP request that is identical to a request in the network cache, CA Access Control automatically permits the request.

You can use the secons utility to display, clean, enable, and disable kernel cache tables.

Protected Resource Tables

When CA Access Control intercepts an authorization request, it checks if the resource to which access is requested is listed in the protected resource tables in the kernel.

If the resource is listed in the protected resource tables, CA Access Control always sends an authorization request to the CA Access Control engine. If the resource is not listed in the protected resource table, CA Access Control may not send an authorization request to the engine but instead resolve the access request in the kernel.

Bypass Tables

When CA Access Control intercepts an authorization request, it checks if the resource to which access is requested is listed in the bypass tables in the kernel.

If the resource is listed in the bypass tables CA Access Control permits the access request. If the resource is not listed in the bypass tables CA Access Control passes the request to the CA Access Control authorization engine for further access checks.

secons -ktc Function—Clean, Enable, or Disable Kernel Cache Tables on UNIX

Valid on UNIX

The secons utility cleans, enables, or disables the kernel cache tables.

This command has the following format:

```
secons -ktc optionNumber
```

-ktc

Specifies to clean, enable, or disable a kernel cache table.

optionNumber

Specifies the action to perform. *optionNumber* must be one of the following:

1

Cleans the F cache table.

2

Enables the F cache table.

3

Disables the F cache table.

4

Cleans the NetwDCM table.

5

Enables the NetwDCM table.

6

Disables the NetwDCM table.

7

Cleans the F inode table.

Note: Not valid on Linux.

8

Enables the F inode table.

Note: Not valid on Linux.

9

Disables the F inode table.

Note: Not valid on Linux.

Example: Clean the F cache Table

The following example cleans the F cache table:

```
secons -ktc 1
```

secons -refIP Function—Refresh IP Addresses for Network Resources

Valid on Windows

The secons utility refreshes the IP addresses of database network resources. For the refresh to work on a particular host, the DNS must have already been refreshed on that host. Use the following Windows command to refresh the DNS manually:

```
ipconfig /flushdns
```

This command has the following format:

```
secons -refIP [hosts]
```

-refIP [*hosts*]

(Windows only) Defines a space-separated list of hosts on which CA Access Control will refresh IP addresses for network resources. If no hosts are listed, local network resources are refreshed.

This option lets you update CA Access Control resources with the current IP address and is particularly useful in a DHCP environment where IP addresses are assigned dynamically.

secons -rl Function—Reload Configuration Settings on UNIX

Valid on UNIX

The secons utility reloads the seos.ini file. This lets you update your configuration settings without having to shut down CA Access Control.

This command has the following format:

```
secons -rl
```

-rl

(UNIX only) Reloads the seos.ini configuration file and updates settings without shutting down CA Access Control.

secons -v Function—Control Instrumentation Run-time Settings on Windows

Valid on Windows

The secons utility controls CA Access Control instrumentation run-time settings. You can use the utility to load an external DLL library into an active process and modify the run-time tracing configuration of CA Access Control instrumentation plug-ins. You must have the ADMIN or OPERATOR attribute to execute this command.

This command has the following format to load a DLL library:

```
secons -v target load "dll_name"
```

This command has the following format to enable or disable the trace on a CA Access Control instrumentation plug-in:

```
secons -v target trace plugin_name
{trace:enable|trace:disable}:{file:"tracefile_path"|debug}
```

Note: CA Access Control does not start the trace until the trace is correctly configured.

This command has the following format to configure the trace on a CA Access Control instrumentation plug-in:

```
secons -v target trace plugin_name trace:option:{sources:{1 | 4} | filtering:value
| filecyclic:{0 | 1} | filelimit:value }
```

debug

Specifies that the command enables or disables tracing to the debug output channel.

file:"*tracefile_path*"

Defines the full path to the file that CA Access Control writes the trace to.

Note: If you specify the trace:disable parameter, CA Access Control ignores any value that you specify for the file:"*tracefile_path*" parameter.

filecyclic:{0 | 1}

Specifies if cyclic file tracing is enabled. If you enable cyclic file tracing, when the size of the trace file reaches the specified maximum size, CA Access Control returns to the start of the trace file and continues writing the trace.

This parameter has the following values:

0-Disable cyclic file tracing

1-Enable cyclic file tracing

filelimit:*value*

Defines the maximum size, in bytes, of the trace file. A value of 0 means the trace file has no maximum size.

filtering:*value*

Defines the bitwise filter mask that filters the trace for the specified instrumentation plug-in. CA Access Control does not write filtered events to the trace file.

Note: To specify no filtering, that is, to specify that CA Access Control writes all events to the trace, use the following value: 0xFFFFFFFF. All other values for this parameter depend on the plug-in that you specify.

load "*dll_name*"

Specifies to load the specified DLL into the target process. The DLL operating environment and the target process operating environment must be identical. For example, if you specify a 32-bit process as the target process, the DLL must also be 32-bit.

Important! The DLL must be located in the *ACInstallDir*\bin folder.

sources:{1 | 4}

Specifies where CA Access Control outputs the trace.

This parameter has the following values:

1-Output to file

4-Output to debug API trace

target

Defines the target process or processes. This parameter has one of the following values:

all_32bit

Specifies to send the command to all 32-bit processes running on the computer.

all_64bit

Specifies to send the command to all 64-bit processes running on the computer.

PID

Defines the process ID of the target process. The target process must be running on the computer.

process_name

Defines a mask that identifies the names of the target process. The target process must be running on the computer. For example, if you specify cmd.exe for this parameter and there are three instances of cmd.exe running on the computer, CA Access Control applies the command to all three processes.

trace *plugin_name*

Specifies to modify the run-time tracing configuration for the CA Access Control instrumentation plug-in named *module_name*, for example, cainstrm or stopplg.

Note: You must specify the DLL name of the plug-in. If you upgrade an instrumentation plug-in and the name of the DLL for the plug-in changes, you must specify the name of the new DLL in the command. For example, if you upgrade the cainstrm plug-in and the name of the upgraded DLL for the plug-in is cainstrm2.dll, you must specify cainstrm2 as *plugin_name*.

trace:disable

Specifies to enable the trace on the target plug-in.

trace:enable

Specifies to disable the trace on the target plug-in.

Note: This parameter changes the status of the trace enabled flag in run time. CA Access Control does not begin the trace until the trace is correctly configured.

trace:option

Specifies to configure the trace on the target plug-in.

Example: Enable Tracing to the Debug Output Channel

The following command changes the status of the trace enabled flag in run time for all files in the stopplg plug-in that are in 32-bit processes running on the computer. CA Access Control does not begin the trace until the trace is correctly configured:

```
secons -v all_32bit trace stopplg trace:enable:debug
```

Example: Apply a Trace Filtering Mask to a Plug-in

The following command applies a trace filtering mask to all files in the cainstrm plug-in that are in the process with PID 362:

```
secons –v 362 trace "cainstrm trace:option:filtering:4294967295"
```

secons -whoami Function—Display Your User Name and Security Credentials

Valid on Windows

The secons utility displays the user name as it is known to the CA Access Control authorization engine. This is the information that it stores in the Accessor Element Entry (ACEE) table. The ACEE stores information about the following users:

- **Logged in user**—A user that has logged in to the operating system. Specific ACEE attributes for this type of user are:
  - Login session ID
  - Login session type
- **Management user**—A user that has logged in to a CA Access Control management application (using an LCA connection). For example, selang.
- **Authorization API user**—A user that was referenced in SEOSROUTE_* API.
- **SPECIALPGM Logical user**—A user that is being references at least in one SPECIALPGM record. A specific ACEE attribute for this type of user is:
  - ACEE association with SPECIALPGM records
- **Built in user**—A user that is built in CA Access Control. For example, *_undefined*.

This command has the following format:

```
secons -whoami
```

Example: Display Your User Name and Security Credentials

This example displays your own user name and security credentials as they are known to the CA Access Control authorization engine:

```
secons -whoami
```

The secons output looks like this:

```
ACEE handle '2' represents 'Logged on User': COMP1-SRV-X86\John
ACEE was created at: Wed Feb 20 17:34:47 2008
ACEE was last accessed at: Wed Feb 20 17:36:49 2008
ACEE user role is: Auditor, Administrator
ACEE audit mode is: Failure, Login Success, Login Failure; Originated from User
definition
ACEE user is a member of 0 'CA Access Control' groups
ACEE's Logon session ID is: 0:68737
ACEE's Logon session type is: Interactive
```

More information:

sewhoami Utility—Display Your CA Access Control User name and Security Credentials on UNIX (see page 220)

secrepsw Utility—Create Policy Model and Shadow Files

Valid on UNIX

The secrepsw utility creates a password record for every user in the /etc/passwd file. This is necessary for administering users defined by PMDBs operating over a UNIX environment. The utility can also create and remove shadow files.

Note: This utility is located in the lbin directory and only root can use it. You must change the shadow token in the pmd.ini file to *yes* before you use this utility.

This command has the following format:

`secrepsw [-h] [-c] [-r PolicyModel] [-s PolicyModel]`

-c

Creates a new Policy Model password file from the /etc/passwd and /etc/shadow files on the local computer.

-h

Displays the help for this utility.

-r *PolicyModel*

Transfers user names and passwords from the Policy Model's shadow file back to the original Policy Model password file (passwd).

-s *PolicyModel*

Transfers user names and passwords from the Policy Model password file (passwd) to the Policy Model's shadow file.

sedbpchk Utility—Back Up the Database

Valid on UNIX

The sedbpchk utility creates a backup copy of the database. It copies the runtime database to a temporary location, performs various database integrity checks on the temporary database, and, if the database passes the checks, copies the temporary database into a backup location.

If the database does not pass the integrity tests, sedbpchk tries to determine whether any updates were applied to the database while the copy was being made. If there were updates, the conclusion that the database is corrupted may not be accurate.

If there were no updates while the database was being copied, the conclusion that the database is corrupted is probably true. In that case, a mail message is sent to the system administrator, who can then use the backup directory to override the corrupted runtime database.

Note: This script is *not* foolproof. It may conclude that a database is corrupted when it is not. However, the conclusion that a database is okay is always accurate.

You must have root and ADMIN privileges to run this script. Before using sedbpchk, we recommend that you review the script, located in *ACInstallDir*/lbin as sedbpchk.sh, to confirm that the values of the following fields match the needs of your site.

MAIL_TO

Specifies the name of the user who is sent the notification that the database is corrupt.

RETRIES

Specifies the number of times the utility checks the database when it suspects that the database is corrupted before sending the notification.

ACInstallDir

Specifies the location of the CA Access Control installation directory.

SE_BINDIR

Specifies the location of the CA Access Control binary files directory.

SE_DB_DIR

Specifies the location of the CA Access Control runtime database directory.

SE_BCKDIR

Specifies the location of the backup database directory.

SE_TMPDIR

Specifies the location of the temporary database directory.

Note: This utility is supplied as a script file; you need to specify the .sh extension to run it.

This command has the following format:

```
sedbpchk
```

seerrlog Utility—Display Error Log Records

Valid on UNIX

The seerrlog utility displays the records in the CA Access Control error log. You must either have permission to read the error log file, or be a member of the group that can read the error log files (the group defined in the configuration setting error_group).

This command has the following format:

```
seerrlog [-h] [-s date] [-e date] [-d] [-f filename]
```

-s *date*

Specifies the start date for the list. Lists records written on and after the defined date.

Limits: Date should be in the format dd-mm-yyyy.

-e *date*

Specifies the end date for the list. Lists records written up to and including the defined date.
Limits: Date should be in the format dd-mm-yyyy.

-d

Specifies *not* to print the detailed information of failures.

-h

Displays the help for this utility.

-f *filename*

Specifies the error log file to read.

By default, seerrlog reads the *ACInstallDir*/log/seos.error file. You cannot define this file in the database, and only CA Access Control can write to the file.

Examples

- To list all error records written since 3 January 2006, specify:

  ```
  seerrlog -s 03-Jan-2006
  ```

- To list all error records written between 3 January 2006 and 1 January 2007, specify:

  ```
  seerrlog -s 03-Jan-2006 -e 01-Jan-2007
  ```

segrace Utility—Display User Login Information

The segrace command line utility displays the number of grace logins left for a user, the number of days remaining until the user's existing password expires, or the date and time the user last logged on, and from which terminal.

Note: For more information about the grace login property of a user, see the *Endpoint Administration Guide* for your OS.

Before segrace can work, the system administrator must activate CA Access Control password checking by entering the selang command:

```
setoptions class+(PASSWORD)
```

Subsequently, every time a user's password is changed, the new password is checked against the password quality rule set in the database.

segrace Utility—Display User Login Settings on UNIX

Valid on UNIX

The segrace utility displays login settings for a user. We recommend that you run the segrace command every time a user logs in. To do so, add the command to /etc/profile and /etc/csh.login (or /etc/.login for Solaris).

To permit segrace to count grace logins, you must use the sepass utility to change passwords. If users have no grace logins left, segrace invokes the sepass utility, which requests that the users replace their passwords. Your site may decide which command to execute instead of the sepass utility by specifying another utility in the sepass_command token in the segrace section of the seos.ini file.

This command has the following format:

```
segrace [-h] [-d days] [-l] [-p] [userName]
```

-d *days*

Displays the number of days that remain until the user's current password expires. The number appears only if the number of days you specify in the *days* parameter is greater than, or equal to, the interval value in the CA Access Control option. If you omit the *days* parameter, segrace uses a default of seven days. This option works only if the user's password was changed using sepass.

-h

Displays the help for this utility.

-l

Displays the date and time the user last logged in, and from which terminal.

-p

Prompts for a new password when a user's password has expired.

userName

If you specify a user name, and the requester has the ADMIN attribute, segrace displays the required login information for the specified user.

If you do not specify a user name, segrace displays the login details for the current user.

More information:

sepass Utility—Set or Replace a Password (see page 182)

segrace Utility—Display User Login Settings on Windows

Valid on Windows

The segrace utility displays login settings for a user. This utility can be executed from a remote machine, as a standalone module.

Note: If you invoke segrace without any parameters, and no grace logins are found for a user, segrace does not display anything.

This command has the following format:

```
segrace [-h] [-d days] [-l] [-p] [-s host] [userName]
```

-d *days*

Sets the warning *days* parameter to be different from the default one configured in the server.

-h

Displays the help for this utility.

-l

Displays the date and time the user last logged in, and from which terminal.

-p

Prompts for a password warning if the password is about to be expired in the *warning days* period and/or if the user has a grace count.

-s *host*

Specifies the remote server name where the CA Access Control database will be used.

userName

If you specify a user name, and have the ADMIN attribute, segrace displays the required data for the specified user.

If you do not specify a user name, segrace displays the login details for the current user.

segracex Utility—Check Password Expiry on UNIX

Valid on UNIX

The segracex utility sets a new password in the X-Windows environment. The segracex utility checks whether the user's password has expired. If it has, segracex displays a window in which the user can replace the password.

The segracex utility is designed to be linked to the user initialization scripts that are invoked after the user logs into the desktop environment.

The utility checks CA Access Control grace login attribute of the user. If the number of remaining grace logins for the user is:

- Zero, segracex forces the user to change the password.
- Positive but less than the value specified in the grace parameter of the user or the global grace setting (if there is one), segracex advises the user to change the password.
- Equal to or greater than the value specified in the grace parameter of the user or the global grace setting (if there is one), segracex does nothing.

When changing the password, segracex prompts the user for the old password. It then prompts the user for the new password.

- If CA Access Control password checking is enabled, segracex checks whether the new password complies with the password rules that are set in the database. If the new password passes the quality check, the user is again prompted for the new password.
- If password checking is disabled, the user is immediately re-prompted for the new password.

  When the new password is entered for the second time, the two copies of the new password are compared. If the copies are not identical, the user is prompted again for the new password.

If the two new passwords are identical, the password is updated in the following ways:

- The local host password files-/etc/passwd and any security files-and the local database are updated.
- If a value is defined in the passwd_pmd or parent_pmd token in the [seos] section of the seos.ini file, the appropriate PMDB is updated, which then propagates the update to its subscribers both in the UNIX environment and the database. If the token nis_env in the [passwd] section of the seos.ini file has a value (either nis or nisplus), the NIS or NIS+ server is updated. When a password is set on a master NIS server, the NIS password map is automatically reconstructed.

The customizable resources, such as colors and fonts, are in the segracex file. During standard installation of CA Access Control, this file is placed in the following directories:

- For all platforms except Sun Solaris:

  `/usr/lib/X11/app-defaults`
- For the Sun Solaris platforms:

  `/usr/lib/openwin/app-defaults`

The icon with the CA Access Control trademark is in the BigTradeMark_BW.xpm file, which you must put into the *ACInstallDir*/data/segracex directory after installation.

This command has the following format:

`segracex [-user` *`userName`*`]`

userName

If you specify a user name, and the requester has the ADMIN attribute, segracex operates on the specified user.

If you do not specify a user name, *segracex* operates on the current user.

SegraceW Utility—Check Password Expiry on Windows

Valid on Windows

This Windows GUI grace utility checks whether the user's password has expired and/or the user has a grace login count. If it has, SegraceW displays a window in which the user can replace the password.

SegraceW can be executed as a standalone module in a non-CA Access Control environment. This enables you to apply this utility on any workstation in a domain.

SegraceW tries to connect first to the primary domain controller (in an NT 4.0 environment), and only if the attempted connection fails, it looks for backup domain controllers. In a Windows 2000 or later environment, SegraceW tries to connect to the first domain controller it finds.

Note: If a remote host is specified explicitly in the SegraceW execution options, then SegraceW connects only to the remote host.

The SegraceW utility is designed to be called from login batch files located at Domain Controller's NETLOGON share.

The SegraceW utility checks whether the user's password has expired and/or the user has a grace login count.

If the grace login count attribute of the user exists, then:

- If the number of remaining grace logins for the user is zero, SegraceW forces the user to change the password.
- If the number of remaining grace logins for the user is positive, SegraceW advises the user to change the password.

If the user does not have a grace login count, SegraceW checks password expiration status.

- If the password is about to be expired in a time frame larger than the value of the *warning days* parameter configured at the server side, SegraceW does nothing.
- If the password is about to expire in a time frame equal or less than the value of the *warning days* parameter configured at the server side, SegraceW advises the user to change the password.
- If the password has been expired, SegraceW forces the user to change the password.

When changing the password, SegraceW displays a ”change password” message that asks the user to provide the old password, the new password, and confirm the new password.

After passing confirmation check, the password is updated in the domain controller's SAM database.

This command has the following format:

```
segracew [d] [-s remoteHost]
```

d

Sets the *warning days* parameter to be different from the default configured in the server.

-s *remoteHost*

Connects to the specified remote host to retrieve information.

Note: Before you can connect to the remote host, copy the encryption library from the remote host to the local host and rename it to defence.dll.

seini Utility—Manage Configuration Files

Valid on UNIX

The seini utility manages CA Access Control database and initialization files for any host. For any host, the seini utility can do the following:

- Display the path of the CA Access Control database
- Display the path of an initialization (.ini) file
- Display the contents of a token from an initialization file
- Set the value of a specific token in a specific section of an initialization file
- Delete a specific token from a specific section of an initialization file

The seini utility also displays all tokens in any of the other .ini files. The name of the initialization file must always end in the suffix *.ini*. You can work on an .ini file from any remote host as long as you have WRITE and ADMIN privileges.

If you do not specify any switch, seini displays the paths of the database and the seos.ini file.

Note: The seini utility can only update the seos.ini file when seosd is *not* running, or when a rule in the database specifically permits it.

seini can perform an intelligent token and section search, by including certain tokens in the seos.ini file. This feature checks for spelling errors by comparing each token or section with the one you specified until it finds an exact or partial match (within a 25% error margin). If it finds the relevant token or section, seini performs the specified operation; otherwise it displays an error message.

Note: The intelligent search feature works only on the host where you invoke the seini utility.

This command has the following format:

```
seini [-d] [host]

seini [-i] [host]

seini [-H host] \
    {[-f [host.]section.token [ini_file]] | \
    [-r [host.]section.token [ini_file]] | \
    [-s [host.]section.token value [ini_file]] | \
    [-sn [host.]section.token value [ini_file]]}
```

-d [*host*]

Displays the path of the database on the remote host. If you do not specify a host, seini displays the path of the local host.

-f [*host*.]*section.token* [*ini_file*]

Displays the value of the token in the section of the specified initialization file on a specified host. If seini cannot find the specified section or token, an empty line appears. You must separate the host, section, and token names with a period (.). If you do not specify the *ini_file*, CA Access Control searches the seos.ini file for the section and token. To display information about the local machine, omit the *host* parameter.

-g *section*

Displays a list of tokens in the defined section.

-h

Displays the help for this utility.

-H [*host*]

Specifies the remote host to be used with the -f, -r, -s, and -sn flags.

-i [*host*]

Displays the pathname of the initialization file seos.ini. If you do not specify a host, seini displays the pathname on the local host.

-r [*host.*]*section.token* [*ini_file*]

Deletes the token from the section of the initialization file in the specified host. If you do not specify the *ini_file*, CA Access Control deletes the token from the seos.ini file.

To delete information on the local machine, specify the section and token names only.

-s [*host.*]*section.token value* [*ini_file*]

Sets the value of the token in the section of the initialization file in the specified host. If you do not specify the *ini_file* parameter, CA Access Control sets the value in the seos.ini file. If the section or token does not exist, and you specified a remote host, CA Access Control creates that section or token.

To create a section or token on the local machine, use the -sn switch.

-sn [*host.*]*section.token newValue* [*ini_file*]

Sets the value of the token in the section of the initialization file in the specified host. If you do not specify the *ini_file* parameter, CA Access Control sets the value in the seos.ini file. If the section or token does not exist, and you specified the local host, CA Access Control creates that section or token.

To create a section or token on a remote machine, use the -s switch.

Examples: Using seini

- To find out where the seos.ini initialization file is located on the local computer, use the following command:

  ```
  seini -i
  ```

- To find out the value of the *trace* configuration setting in the [seosd] section, use the following command:

  ```
  seini -f seosd.trace_file
  ```

- To set the value of the *trace_to* configuration setting in the [seosd] section, use the following command:

  ```
  seini -s seosd.trace_to file
  ```

  The command output should look like this:

  ```
  The token seosd.trace_to now set to file (was file,stop)
  ```

selang Utility—Run the CA Access Control Command Line

The selang utility invokes a command shell that provides access to the CA Access Control database and the native environment. The database is updated dynamically by issuing selang commands from within the command shell.

Note: The result of the command's execution is sent to the standard output unless you include the -o option.

This command has the following format on UNIX:

```
selang [{-c command|-f file}] [{-d path|-p pmdb}] [-o file] [-r file] [-s] \
[-u user pass]
selang [-l] [-o file] [-r file] [-s] [-u user pass]
```

This command has the following format on Windows:

```
selang [{-c command|-f file}] [{-d path|-p pmdb}] [-o file] [-r file] [-s] [-v]
selang [-l] [-o file] [-r file] [-s] [-v]
```

-c *command*

Specifies the selang command to execute. After selang executes the command, it exits.

If *command* contains any spaces, enclose the entire string in quotation marks. For example:

```
selang -c "showusr rosa"
```

-d *path*

Specifies that selang commands update the database in the defined path.

Note: You can only specify a local database.

-f *file*

Specifies that selang commands are read from the defined file rather than from the terminal's standard input.

As selang executes the commands in the input file, the line number of command being executed appears on the screen. The selang prompt does not appear on the screen. After selang executes the commands in *file*, it exits.

-h

Displays the help for this utility.

-l

Specifies that selang updates the default local database, usually *ACInstallDir*/seosdb (where *ACInstallDir* is the directory where you installed CA Access Control).

You do not need to specify this option with -d or -p.

Note: This option replaces selang. It is only valid when seosd is not running, and only an CA Access Control administrator with sufficient native privileges to update the database files can execute it.

-o *file*

Specifies that selang output is written in the specified file. Each time you invoke selang, it creates a new, empty file. If you specify the name of an existing file, selang writes over the information currently in the file.

-p *pmdb*

Specifies that selang commands update the database of the defined PMDB, which must be in the local station (this is the database in the PMDB subdirectory). Changes to the database are not propagated to subscribers.

Note: This option is not valid if either sepmdd or seosd is running on the specified PMDB and is not the same as using the *hosts command*.

Important! Do not make changes that require propagation in this mode. If you use native mode when making updates, CA Access Control updates only the native host files (as defined in the CA Access Control configuration options).

-r *file*

Specifies that selang reads the commands from the defined file. The file should consist of commands in normal selang syntax, separated by semicolons or line breaks. After executing the commands in *file*, selang prompts the user for input.

If you do not define a file for this option, selang uses the .selangrc file in your home directory.

-s

Specifies that selang opens in silent mode, without displaying the copyright message.

-u *user pass*

(UNIX only) Specifies a username and password for running selang.

To use this option, you must set the check_password token in the seos.ini file to yes; this causes CA Access Control to prompt you with “Enter your password” when you run selang -u. You have three attempts to login.

The token no_check_password_users in the [lang] section of the seos.ini file contains a list of users that bypass the password checking during a login to selang.

Note: If the check_password token is set to no (the default), selang does not require any passwords.

-v

(Windows only) Writes command line to output.

Usage notes:

- If -h is used, all other options are ignored.
- You cannot use the -c option with the -f option.
- You cannot use the -d option with the -p option.
- If you specify -d or -p, you do not need to specify -l.

seldapcred Utility—Encrypt and Store a Credential

Valid on UNIX

The seldapcred utility encrypts and stores a credential you provide. This credential is used by LDAP-enabled CA Access Control utilities (such as sebuildla) for retrieving data from an LDAP Directory Information Tree (DIT). Together with the value of the ldap_userdn token in the [seos] section of the seos.ini file, it lets the utility authenticate to the LDAP service. For a simple authentication, the credential is a password corresponding to the ldap_userdn value. For SASL authentication, the credential has different semantics.

The seldapcred utility writes the encrypted credential to *ACInstallDir*/etc/ldapcred.dat

This command has the following format:

`seldapcred [-h] [-w [credential]]`

-h

Displays the help for this utility.

-w [*credential*]

Specifies the credential you want seldapcred to encrypt and store. If you do not provide input to the seldapcred utility, it prompts you to enter this value. By using the interactive mode in this way, you prevent exposing the credential to other users.

More information:

sebuildla Utility—Create a Lookaside Database (see page 104)

seload Utility—Load and Start CA Access Control

Valid on UNIX

The seload utility loads the CA Access Control extension to the UNIX kernel and starts the CA Access Control daemons. The seload utility loads CA Access Control daemons locally and remotely. It also determines whether the CA Access Control extension to the UNIX kernel is loaded on the specified host. If seosd is not running, seload starts the daemon on the specified host. If you omit the -r switch and parameter, the seosd daemon runs on the local host.

You can instruct seload to load one of the following daemons on the remote host: seosd, selogrd, selogrcd, or serevu. This process depends on the tokens.

Use seload if CA Access Control is placed in the boot sequence of the server station.

Notes:

- When CA Access Control is installed, sample initialization files for every operating system supported by CA Access Control are placed in the *ACInstallDir*/samples/system.init directory. Use these files if CA Access Control is to be started as part of the system initialization.
- The seload utility requires that the executable se_loadtest be located in *ACInstallDir*/lbin (where *ACInstallDir* is the installation directory). This program determines whether the CA Access Control extension to the UNIX kernel is loaded.
- When working remotely, the seload utility requires the following:
  - The executable rseloadd is located in CA Access Control dir/lbin. This program runs on the remote host and activates seload.
  - The file /etc/services contains seosload service. You should add this file during CA Access Control installation.
  - The file /etc/inetd.conf contains the rseloadd program. You can add this program during CA Access Control installation.

This command has the following format:

`seload [-c] [-nopmd]  [-r host [`*`daemon`*`]]`

-c

Changes the encryption key that was set using the sechkey -r command.

-nopmd

If you specify the -c switch with the -nopmd switch, seload does not update the Policy Model update file with the new key.

-r *host* [*daemon*]

Loads the seosd daemons, and any other daemon specified in the [daemons] section of the seos.ini file.

If you specify a *daemon*, seload starts only that daemon; it ignores the seos.ini token. You must supply with the daemon's full path.

The seos.ini token in the [daemons] section is used only if you specify a value. It has no default value. If you do specify a value, seload substitutes the value in the token for the standard values of the specified utility or program. For example, if you specify the value selogrd=yes, seload automatically starts the selogrd daemon after it starts the seosd daemon.

selock Utility—Lock the X Terminal Screen

Valid on UNIX

The selock utility protects your X terminal or station whenever you are away from your work area for any length of time. selock supports three modes of operation:

- Monitor Mode
- Saver Mode
- Locked Mode

The default settings of selock combine the saver and lock modes.

Note: For more information about using selock to lock idle stations, see the *Endpoint Administration Guide for UNIX*.

This command has the following format:

```
selock [-delay period] [-display hostname:display#.screen#] [-fodelay factor] \
    [-folevels levels] [-idelay seconds] [-lock-timeout minutes] \
    [-pixmapFile fileName] [-pw-timeout seconds]
```

-delay *period*

Specifies the amount of time the system icon appears at one location on the screen before fading away and moving elsewhere on the screen. This is the standard screen saver activity and prevents screen burn-in. The time period is entered in microseconds.

If you do not define this period, the utility uses the default value of 5000000 (five million).

-display *hostname:display#.screen#*

Specifies which display monitor to lock. You can find the display and screen numbers in an X-session listing of your system. You must have authorization from the user currently running the alternate display monitor defined here.

If you do not define this option, the utility locks your own display.

-fodelay *factor*

Modifies the length of time each fade-out level remains visible on the screen. This lets the user extend the amount of time spent in each step without increasing the number of levels. The default value is 10.

-folevels *levels*

Specifies the number of fade-out steps for the system icon. Increasing the number of fade-out levels causes smoother fading, but the icon takes longer to fade out. By default, the utility uses 20 fade-out steps.

-help

Displays a help screen that explains the various selock options.

-idelay *seconds*

Specifies the amount of time, in seconds, that passes after you log in before monitoring starts. If selock is part of your .login shell, this delay is needed while your system gets organized after you first log in. The default value is 30 seconds.

-lock-timeout *minutes*

If transparent=off, specifies the time, in minutes, selock spends in saver mode before changing to lock mode.

If transparent=on, specifies the time, in minutes, selock spends in monitor mode before changing to lock mode.

The default value, 0 invokes the lock mode immediately, effectively bypassing the saver mode.

-pixmapFile *fileName*

Specifies the XPM file that selock displays in the background when the screen is locked and the transparent=on.

-pw-timeout *seconds*

Specifies the length of time the password dialog box remains on the screen. The default value is 30 seconds. Note that too large a number can cause problems with the X-server. If the password is not entered correctly within the specified period, the password-entry dialog closes and selock remains in lock mode.

-segrace {on|off}

Specifies for selock to invoke segracex after identifying the user and password. However, selock does not invoke segrace if the user ID and password belong to the user whose name appears in the unlocking_user token (located in the [selock] section of the seos.ini). The default value is off.

Note: The segracex utility checks whether the user's password expired; if it has, a dialog appears in which the user can select a new password. For more information, see segrace Utility—Display User Login Settings on UNIX.

-timeout *minutes*

Specifies the period of user inactivity after which selock switches from the monitor mode to the save mode. The default value is 10 minutes.

-transparent {on|off}

Specifies whether selock leaves the contents of the screen visible when in lock mode. If you specify *on*, the display and update of on-going processes continues. To indicate that the screen is locked, selock changes the background by displaying the contents of the file specified with the -pixmapFile option.
The default value is *off*.

-user *user-name*

Specifies the user whose password is prompted for in the password dialog box, when user activity is detected in lock mode. The default value is the current user name. The password of root is accepted, regardless of which user is specified by the user option.

-workhours (*hh*:*mm*-*hh*:*mm*)

Specifies the period in which the user can unlock the screen. Before or after the specified period, the password dialog box does not appear if you touch the keyboard or mouse.

The default value is 00:00-24:00; that is, the user can always unlock the screen.

-xmin *pixels*

Specifies the minimum horizontal distance, in pixels, that the system icon jumps at each move. The default value is 100.

-xmax *pixels*

Specifies the maximum horizontal distance, in pixels, that the system icon jumps at each move. The default value is 300.

-ymin pixels

Specifies the minimum vertical distance, in pixels, that the system icon jumps at each move. The default value is 80.

-ymax pixels

Specifies the maximum vertical distance, in pixels, that the system icon jumps at each move. The default value is 250.

More information:

segrace Utility—Display User Login Settings on UNIX (see page 158)

selockcom Utility—Control the selock Utility

Valid on UNIX

The selockcom utility controls the currently active selock process. This includes restarting and stopping selock, as well as switching between the lock, saver, and monitor modes.

Note: When selock is loaded, it disables the terminal's built-in screen saver to prevent race or overlap conditions between selock and the built-in screen saver. If you stop selock with the selockcom exit switch, no screen saver is active on your terminal. You can restart selock or the terminal's built-in screen saver using the standard X command xset s on. For more information on the xset command, see your UNIX documentation.

This command has the following format:

```
selockcom {-activate|-deactivate|-exit|-restart|-lock} \
    [-display hostname:display#.screen#]
```

-activate

Switches selock from the monitor mode to the saver mode without waiting for the predefined time-out period to pass. The keyboard is locked and the CA Access Control logo appears on the screen.

-deactivate

Switches selock back to the monitor mode. This switch simulates user input to the selock process. If selock is currently in the lock mode, the password dialog appears; enter your password to return to the monitor mode. If selock is in the saver mode, you are returned to the monitor mode.

-exit

Terminates the selock process. You can also terminate selock by sending it a sigterm signal. As a last resort, you can also use the sigkill signal (kill -9). If you use the last method, selock does not exit gracefully; therefore you should not normally use it. If you are running a virtual-root window manager, using kill -9 forces you to restart the window manager to restore the virtual window.

-restart

Terminates the selock process and then immediately restarts it with the same command line options as the previous invocation. This is a good way to get selock to re-read the resource database if the database was changed since you last invoked selock.

-lock

Switches selock to the lock mode, regardless of the current lock-timeout value.

-display *hostname*:*display#.screen#*

Instructs selockcom to control the selock process operating on the specified display. This option allows you to control selock from a remote terminal.

You can find the display and screen numbers in an X-session listing from your system. To do this, you must have authorization from the user currently running the specified display monitor. The default assumption is that you want to lock your own display.

selogmix Utility—Split and Merge Audit Log Files

Valid on UNIX

The selogmix utility splits and merges CA Access Control audit log files.

This command has the following format:

```
selogmix {-s|-m} [-fn fileName] [-l fileName1 fileName2] \
    [-c weight1:weight2] [-t days] [-d] [-i]
```

-c *weight1*:*weight2*

Specifies the correlation of file sizes for splitting files where *weight1* indicates the relative weight of the first file and *weight2* indicates the relative weight of the second file. If you omit this option, selogmix uses a one-to-one correlation.

-d

Specifies to run selogmix in debug mode. In this mode, selogmix displays all settings.

-fn *fileName*

Specifies the name of the audit log file to be split or the resulting file of a merge. If you omit this option, selogmix uses the file name specified by the audit_log token in the [logmgr] section of the seos.ini file.

-h

Displays the help for this utility.

-i

Specifies to run selogmix in interactive mode. In this mode, selogmix prompts you for confirmation before overwriting existing files; otherwise, it overwrites without confirmation.

-l *fileName1 fileName2*

Specifies the files used in the merge or split operation.

You must specify both file names for this option. For merging, specify the two file names you want to merge; for splitting, specify the two destination files. If you omit this option, selogmix uses the file name specified by the audit_log token in the seos.ini file and suffixes 1 and 2 to the file name.

-m

Merges two audit log files.

-s

Splits a specified audit log file.

-t *days*

Specifies a number of days. You can only use this option for splitting files. Specify how many days from the end of logging to put into a separate file. If you omit this option, selogmix separates one last logging day.

Examples

- To split the standard log file into two files of equal size, use the following command:

  ```
  selogmix -s
  ```

  The original audit file is named *ACInstallDir*/log/seos.audit

  The new split files are named *ACInstallDir*/log/seos.audit1 and *ACInstallDir*/log/seos.audit2.

- To separate records for the last two days from the log file, use the following command:

  ```
  selogmix -s -t 2
  ```

- To split a log file into two files with a defined correlation in size, use the following command:

  ```
  selogmix -s -c 1:2
  ```

- To merge two specified files into one named file, use the following command:

  ```
  selogmix -m -l seos.audit1 seos.audit2 -fn seos.audit.merge
  ```

semsgtool Utility—Maintain the Message File

The semsgtool utility lets you:

- Show a single message from the CA Access Control message file
- List an entire section of messages
- Dump the entire file into ASCII files, one ASCII file for each section
- Build a new message file
- Change message to a new one
- List messages, including substring
- Validate the message file

You can only specify one command each time you execute semsgtool.

The default location of the message file is *ACInstallDir*/data/seos.msg

Notes: The CA Access Control message file is comprised of sections and message numbers. Each section holds messages for different CA Access Control modules or sub-modules.

This command has the following format:

```
semsgtool {-build|-b} asciiSourceFile OutputMessageFile

semsgtool {-change|-c} [messageFile] {0xerror-code|section# msg#} new-message

semsgtool {-dump|-d} messageFile

semsgtool {-list|-l} [messageFile] sectionNumber

semsgtool {-number|-n} [messageFile] subString

semsgtool {-show|-s} [messageFile] [0xerror-code|section# msg#]

semsgtool {-validate|-v} [messageFile]
```

-build|-b

Creates a new CA Access Control message file from an ASCII source file.

-number|-n

Lists messages in the message file that include a defined string.

-change|-c

Creates a new message file, named *messageFile*.new, where the specified message has the defined modified string.

-dump|-d

Dumps the message file into several files, one file for each section of the message file. This creates ASCII source files that later can be used to create new CA Access Control message files.

-h

Displays the help for this utility.

-list|-l

Lists all the messages in a given section in the message file.

-show|-s

Shows the message associated with a specific message code.

-validate|-v

(Windows only). Validates the message file by checking for duplicate messages and messages that exceed the allocated boundaries.

0x*error-code*

Defines the hex number of the error code for the message that you want to display or change.

asciiSourceFile

Defines the source file in ASCII format from which semsgtool builds a new message file.

messageFile

Defines the name of the message file. If you omit this option, semsgtool uses the message file as specified in the configuration settings.

OutputMessageFile

Defines the name of a new message file to build.

section# msg#

Defines the section number and message number of the error code for the message that you want to display or change.

sectionNumber

Defines the section number of the section you want to list all the messages for.

Example

- To list the message associated with the error code 0x205, enter the following command:

  ```
  semsgtool -s seos.msg 0x205
  ```

- To list the messages in section 512, enter the following command:

  ```
  semsgtool -l seos.msg 512
  ```

- To create a modified CA Access Control message file, follow these steps:

  1. Create a new message file with a modified message:

     ```
     semsgtool -c 0x2501 "This is the new message"
     ```

     A new message file, seos.msg.new, is created with the modified message.

  2. Copy the new file over the CA Access Control message file:

     ```
     copy seos.msg.new seos.msg
     ```

     Copies the new message file with the modified message on top of the old seos.msg file.

- To show the message associated with the error code 0x0205, enter the following command:

  ```
  semsgtool -s 0x205
  ```

senable Utility—Enable a Disabled User Account

Valid on UNIX

The senable utility enables the login of a user that was disabled for any reason, at any location at which the user was disabled, including PMDBs. For example, a user may have been disabled by the serevu daemon, or because the user's suspend date or expire date arrived.

After enabling the user account, senable calls the sepass utility, which prompts for a new user password. To restore the most recent password, use the -n option.

The senable utility enables an undefined user account by deleting that account from the local /etc/passwd file.

To execute senable remotely, you must explicitly mention your local terminal needs in a rule that grants it WRITE permission for accessing the remote station; otherwise, you cannot perform CA Access Control administration there.

Note: For more information about remote administration restrictions, see the Endpoint Administration *Guide for UNIX*.

This command has the following format:

```
senable [-host hostname] userNames [-n]
```

-host *hostname*

Selects the host with the account to change from disabled to enabled.

You must have ADMIN or PWMANAGER attributes on two hosts to use the -host option:

- The host with the account to be changed from disabled to enabled.
- The host where you enter the senable command.

-h

Displays the help for this utility.

-n

Runs the command non-interactively. If you use this option, senable does not call sepass, and restores the most recently used password.

userNames

Defines a space-separated list of user names for accounts being changed from disabled to enabled.

More information:

serevu Utility—Handle Unsuccessful Login Attempts (see page 208)

senone Utility—Execute a Command as an Unauthorized User

Valid on UNIX

The senone utility executes a command issued by a highly authorized user as an unauthorized user process.

Note: Only highly authorized users who are testing untrusted programs should use this utility.

When you invoke the senone utility, it deletes the process credentials from the authorization daemon. senone then executes a shell with the credentials of a user who is not defined to CA Access Control. From this point on, any program invoked from within this shell is executed with the credentials of the non-CA Access Control user. Because senone does not change the invoker's user ID, the user's UNIX privileges remain unchanged.

Important! We recommend that users who are logged in as root not run untrusted programs. Even when running untrusted programs with senone, unexpected problems can occur.

If you invoke senone without specifying a command, it executes the user's shell as defined in /etc/passwd.

This command has the following format:

senone [*command*]

-h

Displays the help for this utility.

command

Specifies the command you want senone to execute as an unauthorized user.

More information:

sesu Utility—Substitute User (see page 212)
sewhoami Utility—Display Your CA Access Control User name and Security Credentials on UNIX (see page 220)

SEOS_load Utility—Load the CA Access Control Interception Module

Valid on UNIX

The SEOS_load utility controls the dynamic CA Access Control kernel module (SEOS_syscall). The interception module must be loaded before running any CA Access Control utility.

Note: You can use UNIX exits to automatically run programs before and after loading and unloading the kernel.

On streams supported platforms, this utility loads the CA Access Control module to streams depending on the SEOS_use_streams token in the [SEOS_syscall] section of the seos.ini file. If the token is set to yes, the module is pushed into streams.

This command has the following format:

```
SEOS_load [-i|-k|-s|-u]
```

-i

(For HP-UX and Sun Solaris platforms only.) Displays information about the CA Access Control kernel extension.

-k

(For HP-UX and Sun Solaris platforms only.) Loads the CA Access Control module into the kernel without pushing into streams.

-s

(For HP-UX and Sun Solaris platforms only.) Inserts the CA Access Control kernel module into streams. This option ignores the SEOS_use_streams token in the SEOS_syscall section of the seos.ini file.

-u

Unloads the CA Access Control kernel extension from the kernel and then removes the module from streams.

Note: You cannot unload CA Access Control if an application, which is loaded on top of CA Access Control, has an open system call (syscall) that is hooked by CA Access Control. Use *secons -sc* or *secons -scl* to find these processes. You can then shut down these processes and unload the CA Access Control kernel module, or use UNIX exits to automatically shut down these processes before unloading the kernel and then restart them after the kernel unloaded.

sepass Utility—Set or Replace a Password

Valid on UNIX

The sepass utility sets a new password or replaces an existing password in the local host, in a Policy Model, or in the NIS or NIS+ server, as applicable.

The sepass utility changes the user password. Additionally, privileged users can use sepass to change the passwords of other users. When changing your own password, sepass prompts you for your old password.

Note: If seosd is not running, sepass runs a default password program. The DefaultPasswdCmd token in the passwd section of the seos.ini file specifies the default password program. Passwords are stored and transferred over the network in an encrypted format.

This command has the following format:

```
sepass [-d] [-l] [-p] [-s policy_model@hostname] \
    [-g number] [-x] [userName]
```

-d

Displays all the information it has regarding the password update, such as on which stations the update succeeded and if you did not activate setoptions class+(PASSWORD), that the password's quality was not checked. This switch is useful when debugging.

-g *number*

Defines the number of grace logins for *userName*.

-h

Displays the help for this utility.

-l

Replaces the password only on the local station; that is, in the local password file (usually /etc/passwd), security files, and the local database.

In the NIS/NIS+ environments, users are not usually defined in the /etc/passwd file of the client; therefore, the password on the client station is not updated.

In NIS/NIS+ server stations, the password is updated locally and propagated by NIS/NIS+.

This switch and the -p and -s switches are mutually exclusive.

-p

Changes the password only on the remote station and on the PMDB at the host specified in the switch. This switch and the -l and -s switches are mutually exclusive.

-s *policy_model@hostname*

Replaces the password on the local station and on the PMDB at the host specified in the switch. This switch and the -l and -p switches are mutually exclusive.

-x

Replaces the password as if changed by the user *username*. This switch updates the time and date of the last change in the database. Grace logins are terminated.

Note: To let you change the root password as if changed by root, you have to set the RootPwAsOwn appropriately. For more information about seos.ini tokens, see the *Reference Guide*.

username

(Optional) Specifies the name of the user whose password sepass changes. If you omit this option, your own password is set.

Examples

The following examples illustrate how you can use sepass in a variety of situations:

- To change your own password on the local host, enter the command:

  ```
  sepass -l
  ```

  Note: If no PMDB is defined at the site, you can omit the -l switch. If a PMDB is in use at the site, omitting the -l switch changes your password on all subscriber databases of the PMDB. In an NIS/NIS+ client, this switch *does not* change the password; in an NIS/NIS+ server, the password is changed and then propagated.

- To change the password of any user other than your own, on the local host only, enter the command:

  ```
  sepass -l username
  ```

  username must exist in the /etc/passwd file, the appropriate UNIX security files, and the database.

  In an NIS/NIS+ client, sepass does not change the password. In an NIS/NIS+ server, the password is changed and then propagated.

- To change the password of a user on several stations at a site where NIS is not in use, follow these steps:

  1. Create a PMDB.

     Note: For more information about creating PMDBs, see the *Endpoint Administration Guide for UNIX*.

  2. Add all the users whose details must be distributed to the subscriber computers, to both the UNIX and the CA Access Control environments of the PMDB.

  3. Subscribe all the stations to receive the updated passwords to the PMDB.

  4. On every subscriber, set the tokens in the [seos] section of the seos.ini file to the names of your PMDB. For example:

     ```
     passwd_pmd = PMD1@morocco
     parent_pmd = PMD1@casablanca
     ```

  5. Enter the command:
     `sepass username`

     When sepass completes execution, the user's password is changed on all the subscriber databases.

sepmd Utility

The sepmd utility is the Policy Model management utility.

It lets you perform the following tasks:

- Administer subscribers and the update file
- Administer Dual Control
- Manage the Policy Model log file
- Manage the PMDB
- Backup the PMDB
- Restore the PMDB

Note: You must run the sepmd utility on the host where the Policy Model resides.

More information:

sepmd Utility—Administer Dual Control (see page 189)
sepmd Utility—Back Up the PMDB (see page 191)
sepmd Utility—Manage the Policy Model Log File (see page 192)
sepmd Utility—Manage the PMDB (see page 193)
sepmd Utility—Restore the PMDB (see page 195)

sepmd Utility—Administer Subscribers and the Update File

The sepmd utility creates, removes, and assigns subscribers.

This command has the following format:

```
sepmd {-C|-de|-l|-L|-p|-R} pmd

sepmd {-n|-r|-u} pmd subscriber

sepmd -s pmd subscriber offset

sepmd -sm pmd mf_subscriber mf_type mf_sysid mf_admin offset

sepmd -smq pmd <-predefined> <ACMQ queue> [-destination <destination>}

sepmd -t pmd {auto|offset}
```

-C

Displays all commands in the update file, and their offsets. The offset indicates the location of the update inside the file, which, you may want to specify when you subscribe another database or PMDB.

-de

(UNIX only) Decrypts the information in the encrypted updates.dat file. Data encryption for this file occurs when you set the UseEncryption PMDB configuration setting to yes.

-l

Lists the subscribers of the Policy Model.

-L

Lists the Policy Model and its status, including number of errors, availability, offset, synchronization mode, and the next command to be propagated. The update file contains all updates that must be, or have been, propagated by the Policy Model. The offset indicates the location of the next update that must be sent to a subscriber. Both initial and latest offsets also appear.

-n

Creates a new subscriber and then updates it retroactively to the Policy Model. For general rules that apply for updating a subscriber, see the description for the -s option.

Note: This option sends the contents of the entire PMDB-including the LOGINAPPL (UNIX only) and SPECIALPGM objects-to the new subscriber. You may want to filter out these objects if the subscriber's objects differ from those of the parent.

The -n option does not replace the Policy Model database definitions on the target subscriber database definition, rather it is added to the existing Policy Model. If the target database contains additional resources or attributes, the new Policy Model does not remove them after subscription is complete.

A subscriber added with -n is marked as *sync*, indicating that it is now in synchronization mode and receives all of the PMDB rules. When the subscriber has received all the rules, it is released from synchronization mode and becomes a regular subscriber. The -n option may take some time to process. If there are multiple or contradictory updates, the last one is used.

Important! When you subscribe a CA Access Control endpoint or a PMDB to another PMDB using *sepmd -n*, the new parent PMDB should not contain any policies (POLICY object names) that already exist in the new subscriber. Undeploy each existing policy from the subscriber and then delete the POLICY object and linked RULESET object from the subscriber before you subscribe it to the new parent PMDB.

On UNIX, if the send_unix_env token in the seos.ini file is set to yes, the -n option also sends the contents of Policy Model password and group files. We recommended that you view the database, by using dbmgr -export -l, to ascertain the commands being forwarded.

-p

Lists the resident Policy Models and their status.

-r

Removes the subscriber from the list of unavailable subscribers maintained by sepmdd, making the subscriber available for immediate updates. Normally, if a subscriber is down and cannot receive updates from the Policy Model, sepmdd tries to send updates to that subscriber only after a certain period of time. However, if you specify this option, sepmdd skips the waiting period and tries to send updates to the subscriber immediately.

-R

Update all subscribers with their real offset.

-s

Subscribes another database or PMDB to the Policy Model. When you subscribe a host to a Policy Model, the host must be up, and CA Access Control must be running on that host. Additionally, the PMDB must be the parent PMDB of the subscribed host. You establish this relationship with the parent_pmd subscriber's configuration setting, which must contain the name of the PMDB to which the host is being subscribed.

When you subscribe a Policy Model to another Policy Model,

- the token parent_pmd in the pmd.ini file of the subscribed Policy Model must contain the name of the Policy Model to which it is subscribing (its parent Policy Model).
- CA Access Control must be running on the host in which the subscribed policy resides.

A PMDB should have only one parent. If you decide to establish a Policy Model with more than one parent give the parent_pmd token the name of a file containing a list of the parent Policy Models. However, establishing more than one parent is not recommended because you risk inundating your database with unreliable instructions from multiple sources.

-sm

Assigns a mainframe subscriber to the Policy Model.

-smq

Subscribes a pre-defined message queue subscriber to a policy model.

<ACMQ queue>

Specifies the following pre-defined Message Queue queues:

- ServerToServer
- ServerToServerBroadcast
- ServerToEndpointBroadcast
- EndpointToServer
- ServeryoEndpoint

-destination

Specifies the destination of the CA Access Control component that receives messages from the subscriber.

-t

Truncates the update file by deleting entries from it.

Note: On UNIX, if the force_auto_truncate PMDB configuration setting is set to no, sepmd -t does not truncate the update file. If the token is set to yes, the command truncates the update file even if there are no subscribers to the Policy Model.

- If you are using *offset* (manual cutting), you can find the offset by running sepmd with the -L option.

  Note: You must use the true offset provided in the -L parameter to truncate the file, and not an offset derived by subtracting from the start offset.

- If you are using auto, sepmd calculates the offset of the first unpropagated entry and deletes all the entries before it. Using auto saves the step of running the utility with the -L parameter.

If a subscriber received fewer than all updates before the specified offset, sepmd displays an error message and does not truncate the file. If you want to truncate the file anyway, do the following:

- Unsubscribe the host that was not updated
- Truncate the file
- Resubscribe the host to the Policy Model

If you do this, the subscriber fails to receive one or more updates from the Policy Model. The subscriber's offset changes to the last offset of the updates file.

-u

Removes a subscriber from the Policy Model subscription list.

auto

Instructs sepmd to calculate the offset of the first unpropagated entry and to delete all the entries before it.

offset

Used with the -s or -sm options, specifies the point within the update file from where the newly added subscriber starts receiving updates.

Used with the -t option, specifies the distance from the beginning of the update file to the position of a particular subscriber.

Use the -C option to see the valid update offsets. If you specify an offset that is in the middle of an update, the offset is moved forward to the beginning of the next update. If you specify an invalid offset (smaller than the first offset or larger than the last), an error message appears.

pmd

Specifies the name of the Policy Model.

-predefined

Specifies to use pre-defined message queue subscribers

subscriber

Specifies the subscriber station or the host of the subscriber PMDB.

sepmd Utility—Administer Dual Control

Valid on UNIX

The sepmd utility manages Dual Control transactions. The sepmd utility gives each transaction a unique ID number when it is created.

Note: For more information about Dual Control, see the *Endpoint Administration Guide for UNIX*.

When you use Dual Control, the name of the PMDB must be *maker* and the is_maker_checker configuration setting must have the value yes for both the PMDB and CA Access Control.

This command has the following format:

```
sepmd -m {l|la|lo}
```

```
sepmd -m {d|r} transactionId
```

```
sepmd -m p transactionId code
```

-m d

Deletes the transaction. A transaction is one or more commands that must be approved before they are implemented on the PMDB. Only the user who created the transaction can delete it.

-m l

Lists the unprocessed transactions (awaiting the Checker) of the user who invoked the command. Each transaction is listed with its ID number, the name of its Maker (the user who created the transaction-in this case the same user who invoked the command), and its description, if any.

-m la

Lists all the unprocessed transactions of all the Makers. Each transaction is listed with its ID number, the name of its Maker, and its description, if any.

-m lo

Lists the unprocessed transactions (awaiting the checker) of all the Makers *except* the transactions of the user who invoked the command.

-m p

Processes a transaction. When the Checker (any admin user *except* the Maker who created the transaction) enters an ID number, all the commands in the specified transaction appear in a list.

This option does not work in the following circumstances:

- If one or more of the commands in the transaction pertain to the user who invoked the command.
- If the transaction is locked by a different Checker.
- If the transaction was created by the user who invoked the command-Makers cannot act as Checkers for their own transactions.
- If the specified transaction ID does not exist.
- If the user who invokes the command does not have the authority to be a Checker.

-m r

Retrieves or locks a transaction.

- If you are the user who created the transaction (the Maker) this parameter retrieves a specific, unprocessed transaction. After you retrieve the transaction, you can direct it to an appropriate file and use the ASCII editor of your choice (vi, emacs, and so on) to update the transaction.
- If you are a user who is *not* the Maker (Checker) this parameter locks the transaction prior to processing. You cannot change a locked transaction.

transacationID

Specifies the unique identifying number that sepmdd gives to the transaction when it is created.

code

Specifies a numeric code that specifies what the Checker should do when processing the transaction:

0

Rejects the transaction, in which case all the commands in the transaction are deleted and no changes are implemented in the PMDB

1

Authorizes the transaction, in which case the commands are immediately implemented in the PMDB

2

Unlocks the transaction so that it can be processed later, or by a different Checker.

sepmd Utility—Back Up the PMDB

The sepmd utility lets you back up the Policy Model database.

This command has the following format:

```
sepmd {-bl|-ul} pmd
```

```
sepmd -bd pmd destination
```

```
sepmd -bh pmd destination backup_host
```

-bd

Backs up *pmd* to the directory *destination*.

-bh

Backs up *pmd* to the directory *destination* for Policy Models in a hierarchy. That is, the backup modifies the PMDB subscribers so that the subscription still works when the backup is moved to the *backup_host* host.

-bl

Locks the *pmd* so that it does not propagate commands to subscribers.

Use this if the Policy Model has subscribers and you want to make sure updates are not accepted while the backup is in process.

-ul

Unlocks a locked *pmd*.

backup_host

Defines the name of the host where you intend to move the backup host to.

destination

Defines the name of the directory where you want the PMDB files to be backed up to.

pmd

Defines the Policy Model database, which is located in the location specified by the _pmd_directory_ configuration setting.

Example: Back Up a PMDB

The following command back up a PMDB named myPMDB to the /tmp/my_pmdb directory:

```
sepmd -bd pmdb /tmp/my_pmdb
```

You can now manage the PMDB as required:

```
selang -d /tmp/my_pmdb
```

Example: Back Up a PMDB with Subscribers

The following commands show you how to back up a PMDB that has subscribers and then move the PMDB to a different host:

1. Lock the PMDB:

   ```
   sepmd -bl mainPMDB
   ```

   CA Access Control locks the PMDB so that it does not send or receive updates.

2. Back up the PMDB:

   ```
   sepmd -bh mainPMDB /tmp/my_pmdb host63
   ```

   CA Access Control backs up the PMDB to the /tmp/my_pmdb

   On UNIX, CA Access Control updates subscribers.dat with the backup host name you specified.

   On Windows, CA Access Control creates a *pmd*.reg file, which is a dump of the *pmd* registry settings with the Parent_Pmd configuration setting value changed to match the new host you specified.

3. Unlock the PMDB:

   ```
   sepmd -ul mainPMDB
   ```

   CA Access Control unlocks the PMDB.

4. Transfer the PMDB backup to its new host.

   Note: The new host must have the same OS and CA Access Control version as the current computer.

5. (Windows only). Import the mainPMDB.reg file into the registry on the new host.

   You can now continue to use the PMDB as you normally would.

sepmd Utility—Manage the Policy Model Log File

The sepmd utility manages the Policy Model log file. The Policy Model log file provides a detailed audit trail of Policy Model data base activities. For example:

```
Wed Nov  4 10:08:02 2003 pmdb1:Processing list request for missouri.yourco.com
Wed Nov  4 10:08:02 2003 pmdb1:Processing list request for oregon.yourco.com
Wed Nov  4 10:09:14 2003 pmdb1:Empty request
Wed Nov  4 10:09:15 2003 pmdb1:Processing shutdown request
Wed Nov  4 10:09:15 2003 pmdb1:Delete filters
Wed Nov  4 10:10:04 2003 pmdb1:Opened error logs
Wed Nov  4 10:10:04 2003 pmdb1:Try to load filters
Wed Nov  4 10:10:04 2003 pmdb1:Filters file : nis_filter.dat
```

Running sepmdd for the first time automatically creates the Policy Model log file.

On UNIX, you can use the pmd_log_level PMDB configuration setting to control what the PMDB logs:

- **0** - Do not log any entries.
- **1** - List only error messages.
- **2** - List error and informational messages (default value).

Note: A warning message in the log file tells you if you have exceeded file size limitations. Use configuration settings to increase the size if the log file.

This command has the following format:

```
sepmd {-sl|-kl|-dl|-cl} pmd
```

-cl

Clears the contents of the Policy Model log file.

-dl

Displays the Policy Model log file.

-kl

Makes the Policy Model log file unavailable.

-sl

Makes the Policy Model log file available.

pmd

Specifies the name of the Policy Model.

sepmd Utility—Manage the PMDB

The sepmd utility stops and starts Policy Models and on UNIX, it also reloads configuration settings that affect the Policy Model.

Note: In Windows, unlike UNIX, sepmd does not stop or start the Policy Model service. Instead, it activates and deactivates the Policy Model.

You must have ADMIN authority in the Policy Model to use sepmd for starting or querying the Policy Model.

This command has the following format:

```
sepmd {-c|-e|-k|-S} pmd
```

```
sepmd -tm seconds
```

-c

Clears the Policy Model error log.

-e

Displays the Policy Model error log.

-k

On UNIX, shuts down the Policy Model daemon safely. On Windows, it deactivates the Policy Model service.

Note: Do not use the kill command on UNIX to shut down the Policy Model daemon.

-ri

On UNIX, it reloads the Policy Model and CA Access Control configuration files (pmd.ini and seos.ini respectively) while sepmdd is running. You can only use this option at intervals of one minute or more. This option checks configuration changes in the following tokens: parent_pmd, _retry_timeout_, _min_retries_, and _shutoff_time_.

On Windows it reloads Policy Model information from the registry to the hosts. Use this if you changed data and want to be sure it is sent to the host PMDBs.

-S

On UNIX, starts the Policy Model daemon. On Windows, it activates the Policy Model service.

Use this option to start the daemon when you do not have any other commands to execute.

-tm *seconds*

(Windows only) Sets an initial timeout interval (in seconds) for executed request.

pmd

Specifies the name of the Policy Model.

sepmd Utility—Restore the PMDB

The sepmd restores a PMDB on a local host. The backup files you use to restore the PMDB must be from a host running the same platform, operating system, and version of CA Access Control as the restoration host. CA Access Control must be running on the restoration host.

Note: If you back up and restore the PMDB on different terminals, the PMDB does not automatically update the terminal resource in the restored PMDB database. You must add the new terminal resource to the restored PMDB. To add the new terminal resource, stop the restored PMDB, run the *selang -p pmdb* command, then start the restored PMDB.

This command has the following format:

```
sepmd -restore pmd [-source path] [-admins user[,user...]]\
[-xadmins user[,user...]] [-parent_pmd name[,name...]]
```

-restore

Restores the PMDB on the localhost.

-admins *user*[,*user*...]

(UNIX) Defines internal users as administrators of the restored PMDB.

-parent_pmd *name*[,*name*...]

(Optional) Defines the name of the restored PMDB's parent PMDBs. Specify the parent PMDB name in the format *pmdb@host*.

pmd

Defines the name of the PMDB to restore.

-source(*path*)

(Optional) Defines the directory where the backup files are located. If you do not specify the source directory, the PMDB is restored from the files in the default location. The default location is defined in the *_pmd_backup_directory_* token.

Default: (UNIX) *ACInstallDir*/data/policies_backup/pmdName

Default: (Windows) *ACInstallDir*\data\policies_backup\pmdName

-xadmins *user*[,*user*...]

(UNIX) Defines enterprise users as administrators of the restored PMDB.

sepmdadm Utility—Create PMDB Definitions

Valid on UNIX

The sepmdadm utility creates the definitions needed to run a PMDB. The sepmdadm utility is a script consisting of the CA Access Control and UNIX commands required to define a PMBD, to define the relationship of the PMDB to PMDBs above and below it, and to define its subscriber stations. By default, the user root is defined as the administrator and auditor of the PMDB. You must run the sepmdadm utility locally, although you can also run it through a remote shell. When you use sepmdadm to create a new PMDB, you probably want to follow up by pointing subscribers to the PMDB and by synchronizing the UIDs and GIDs.

You can run this utility in interactive or non-interactive modes:

- In non-interactive mode, you enter arguments in the command line. The utility builds the PMDB and its hierarchy according to the values it receives.
- In interactive mode, you do not enter arguments in the command line. The sepmdadm utility asks the user if the desired mode is interactive. If the user answers “y,” then the utility proceeds to ask the user for option values.

When creating a new PMDB with sepmdadm, you identify the stations that are the subscribers of the Policy Model. However, you must also update the parent_pmd token in each subscriber's seos.ini file with the name of the PMDB to which you have subscribed the station. If you do not do this, the subscribers do not accept updates from the PMDB.

By subscribing several stations to the same PMDB, and by subscribing one PMDB station to another, you can create a hierarchy of PMDBs.

This command has the following format:

`sepmdadm` *`options`*

--admin *name*

Defines the CA Access Control administrator of the PMDB.

--auditor *name*

Defines the CA Access Control auditor of the PMDB.

-c | --clean *pmdbName*

Removes the specified Policy Model. This option shuts down the Policy Model daemon, removes the file protections from the database, and deletes the Policy Model directory with all its contents.

You cannot use this option with the --noconfirm option.

--desktop *hostname*

Specifies a station from which the administrators can administer PMDBs located on the local host. If you do not specify any stations, the administrators can only administer the PMDBs from the local host.

--group_fname *fileName*

Defines the location of the groups file under NIS.

-h | --help

Displays the help screen.

-i | --interactive

Runs sepmdadm in interactive mode.

-l

Specifies to run sepmdadm in local mode, meaning that you can create a PMDB when CA Access Control is not running.

Note: Unless you specify this option you must have CA Access Control running to use sepmdadm.

--nis | --NIS

Performs NIS setup on the Policy Model. You must use this option if the PMDB is installed on a NIS server.

--noconfirm

Specifies that the user is not asked to confirm answers. This option is useful when invoking sepmdadm from within a shell script in non-interactive mode.

--parentpmd *pmdbName*

Specifies the name of the parent PMDB to which this PMDB is subscribed. If you use this parameter with the -subsconfig parameter, sepmdadm updates the parent_pmd token in the seos.ini file. If you use this parameter without the --subsconfig parameter, sepmdadm updates the parent_pmd token in the pmd.ini file.

Note: If you want to define multiple parent Policy Models, you must to use quotation marks. For example, to create a Policy Model and define its parent, use the following command:

```
sepmdadm --pmdname subs2 --admin abc123 --admin root --auditors abc123 --desktop
pcp36949 \
--parentpmd "aa@pcp36949,bb@pcp36949"
```

--passwd_fname *fileName*

Defines the location of the passwd file under NIS.

--passwdpmd *pmdbName*

Specifies the PMDB to which sepass sends password updates. This option updates the passwd_pmd token in the [seos] section of the seos.ini file.

Note: You can use this parameter only when you also use the --subsconfig switch.

When creating a multi-level Policy Model, set this parameter to the PMDB at the top of the pyramid, so that password changes can be propagated to all levels in the PMDB system.

--pmdname *pmdbName*

Specifies the name of the PMDB to be created.

--pwmanager *name*

Specifies the CA Access Control password manager of the PMDB.

--seosdir *directory*

Specifies the directory in which CA Access Control is installed. Use this option only if CA Access Control is not installed in the default directory.

--subsconfig

Specifies that the local station is a subscriber. When using this parameter, you must specify the parameters --parentpmd *pmdbName* and --passwdpmd *pmdbName* to update the relevant tokens in the seos.ini file.

Note: The parameters should follow the -subsconfig option when configuring a subscriber.

--subscriber *name*

Specifies subscribers of this PMDB. They can be PMDBs or stations.

--xadmin *name*

Defines the enterprise user administrator of the PMDB.

--xauditor *name*

Defines the enterprise user auditor of the PMDB.

--xpwmanager *name*

Specifies the enterprise user password manager of the PMDB.

Example: Create a PMDB using the command line

Suppose you have a station called bigcentral, where you want to maintain a PMDB for other stations to subscribe to. To create the PMDB at bigcentral, run sepmdadm there. This utility is located in the directory *ACInstallDir*/bin.

To create a PMDB on bigcentral named pmdb1 with workstat1 and workstat2 as subscribers and enterprise users adm1 and adm2 as administrators, run the following command from bigcentral:

```
sepmdadm --pmdname pmdb1 --subscriber workstat1 --subscriber workstat2 \
    --xadmin adm1 --xadmin adm2
```

Example: Pointing subscriber stations to the PMDB

To establish a station as a subscriber to a PMDB, it is not sufficient to specify the subscriber's name at the PMDB's station; you must also perform a procedure at the subscriber station.

To subscribe the local station to a PMDB using the command line, you must use the parameters --parentpmd and --passwdpmd, in addition to the parameter --subsconfig.

For example, to subscribe the local station to the PMDB called pmdb2 located on HOST2 and to the password PMDB called master1 located on HOST1, enter the following command:

```
sepmdadm --subsconfig --parentpmd pmdb2@HOST2 --passwdpmd master1@HOST1
```

sepropadm Utility—Administer Database Properties

The sepropadm utility adds, updates, and deletes properties in the database. Invoke this utility from the directory in which the database resides, and while the CA Access Control is *not* running. The sepropadm utility can add only one property at a time.

Important! This utility is for CA Access Control technical support personnel use only. Do *not* use sepropadm with a description file that was *not* certified by CA Access Control technical support personnel.

This command has the following format:

```
sepropadm file
```

file

Specifies a description file supplied by CA Access Control support personnel. The description file uses the following format:

- There must be one line that begins with the hash symbol (#); it must precede the description lines.
- Lines that begin with semicolon(;) are comments and are not processed.
- The description line to add a new double link OID must conform to the following format:

  ```
  CLASS=%s    PROPERTY=%s    TYPE=%d    SIZE=%d    FLAGES=%x
  ```

- The description line to add a new property must conform to the following format:

  ```
  CLASS=%s  PROPERTY=%s  TYPE=%d  SIZE=%d  FLAGS=%x  LINK2CLASS=%s
  ```

- The description line to delete a property must conform to the following format:

  ```
  CLASS=%s  PROPERTY=%s
  ```

- The description line to change a property must conform to the following format:

  ```
  CLASS=%s  PROPERTY=%s  TYPE=%d  SIZE=%d  FLAGES=%x REPLACE=YES
  ```

Example: A description file for sepropadm

```
The following is a sample description file.
 ; Sample Patch File for the CA Access Control database
 ; Copyright 2004 Computer Associates International, Inc.
 ; ------------------------------------------
 ; DO NOT USE THIS FILE UNLESS YOU KNOW HOW TO!
 # seclassadm database add property patch utility
 ; Format is :
 CLASS=PROGRAM PROPERTY=MD5 TYPE=31 SIZE=16 FLAGS=0
```

sepurgdb Utility—Purge Database References to Undefined Records

Valid on UNIX

The sepurgdb utility searches the entire database for references to undefined records, and then deletes those references from the database, thereby reducing the size of the database.

Important! For safety purposes, first back up the database, and then invoke the utility while the CA Access Control daemons are *not* running.

When a record is deleted, references to it in lists such as ACLs or lists of group membership are usually left as is, to reduce processing time. This does not cause any problems, since CA Access Control assigns a previously unused, unique ID to each new record. You only need to use this utility to to free up some disk space.

To run sepurgdb, you must be root and invoke the utility from the directory containing the database files. The database management system uses pre-allocated disk space. The size of the database file normally does not change significantly after purging. When the size of the database is increased later, the file size may not change significantly due to the pre-allocation.

This command has the following format:

```
sepurgdb FilePath [Username]
```

FilePath

Specifies the base name for the utility's log files. The sepurgdb utility creates two log files:

***FilePath*.err**

Contains a log of errors encountered.

***FilePath*.log**

Contains a log of actions taken.

Note: You can merge the two logs and direct them to the standard output by specifying a minus sign (-) for *FilePath*.

Username

(Optional) Specifies the name of the user that sepurgdb uses to replace deleted owners (users that no longer exist) of the group connection for the USER record.

Note: The user you define must exist in the database, otherwise the utility ignores this option.

sereport Utility Reports Configuration

The sereport utility provides HTML reports, accessible from a web browser, of database and Policy Model information. sereport operates on the current database used by the authorization engine.

You can set sereport options for the utility:

- On UNIX, sereport uses a configuration file that you specify using the -f option.

  By default, this is *ACInstallDir*/etc/sereport.cfg

- On Windows, sereport uses the registry, which you can configure. The registry settings for sereport are defined under the following key:

  ```
  HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Report
  ```

The reports you can generate, their description and corresponding configuration file settings or registry keys are shown in following table.

Report Number	Title and Description	Section\Subkey	Tokens\Entries
1	Administrative Privileges Display specified administrative privileges of users.	admin_report	■ Hostname ■ Objects_Pattern ■ User_Mode
2	Login Limitation Display login limitations of users.	disablelogins_report	■ Hostname ■ Objects_Pattern ■ Properties ■ User_Mode
3	Dormant Accounts Display inactive accounts by date (days). If an account does not have any login information, the create time is used to calculate dormant days.	dormant_report	■ Hostname ■ Objects_Pattern ■ Dormant_account ■ User_Mode
4	Last Login Display last login date of users.	login_report	■ Hostname ■ Objects_Pattern ■ User_Mode
5	Password Change Display list of users whose passwords must be changed within the specified number of days.	passwd_report	■ Days_to_change ■ Hostname ■ Objects_Pattern ■ User_Mode
6	Warning Mode Display resources with objects in warning mode.	warning_report	■ Class_Name ■ Hostname ■ Objects_Pattern
7	Untrusted Programs Display programs in untrusted mode.	untrust_report	■ Hostname ■ Objects_Pattern
8	Users' Privilege Access Rights Show access privileges of users to specified resources.	accessor_report	■ Accessor ■ Class_Name ■ Hostname ■ Objects_Pattern

Report Number	Title and Description	Section\Subkey	Tokens\Entries
9	Compare users/groups in databases Display users and groups that are defined in some but not all, databases.	grp_usr_compare	■ Hostname ■ Objects_Pattern
10	Compare Protected Resources Display whether resources are defined in the specified databases.	res_compare	■ Class_Name ■ Hostname ■ Objects_Pattern
11	Compare Access Rights Display the differences in resource restrictions between a Policy Model and a subscriber database.	acc_compare	■ Class_Name ■ Hostname ■ Objects_Pattern
12	Compare Users' Information Display differences in user definitions between a Policy Model and a subscriber database.	usr_compare	■ Hostname ■ Objects_Pattern ■ Properties
13	Compare PMDB and Subscriber Display the rules (as defined by the Class_Name and Object_pattern tokens) that exist on the PMDB, but do not exist on the subscriber database. Note: If all of the rules on the PMDB exist on the subscriber database, then the databases are reported as IDENTICAL.	pmdb_compare	■ Class_Name ■ Hostname ■ Objects_Pattern

Accessor

Specifies the pattern (mask) for accessor selection. Use * to select all accessors.

Class_Name

Specifies a list of classes.

Days_to_change

Specifies the number of days left until the user is requested to change passwords.

Dormant_account

Specifies the period the account is to be considered dormant.

Hostname

Specifies a list of hosts from which the data is retrieved.

Objects_pattern

Specifies the pattern (mask) for object selection. Use * to select all objects.

Properties

Specifies attributes associated with the objects.

Report_place

(UNIX only) Specifies the full path location where the report is printed.

Note: On Windows, you define the location of the output using the -f option of the command.

User_Mode

Specifies a comma-separated list of user modes.

You can also find the following additional configuration settings in the colors section\key:

title

Specifies the color of the report's title.

class_title

Specifies color of the report's class_title.

background

(UNIX only) Specifies the color of the title report's background. The background and logo must be written in full path.

logo

Creates the logo. The background and logo must be written in full path.

sereport Utility—Create HTML Reports on UNIX

Valid on UNIX

The sereport utility creates HTML reports, accessible from a web browser, of database and Policy Model information. sereport operates on the current database used by the authorization engine.

To use sereport, you need READ privileges in all queried databases.

Note: The default configuration file is *ACInstallDir*/etc/sereport.cfg

This command has the following format:

```
sereport [-f|-file pathname] -r|-report number [-host hostnames]
```

-f | -file *pathname*

(Optional) Specifies the full path of the configuration file. If you do not specify a file, sereport uses the default file *ACInstallDir*/etc/sereport.cfg

-host *hostnames*

(Optional) Specifies the names of one or more hosts you want to report on. If you do not specify a host, sereport takes the host from the config file.

-r | report *number*

Specifies the report number to create.

sereport Utility—Create HTML Reports on Windows

Valid on Windows

The sereport utility creates HTML reports, accessible from a web browser, of database and Policy Model information. sereport operates on the current database used by the authorization engine.

To use sereport, you need READ privileges in all queried databases.

This command has the following format:

```
sereport -f|-file pathname -r|-report number [-host hostnames]
```

-f | -file *pathname*

Specifies the full pathname of the output file (the report).

Note: The content of the specified file is structured in HTML format so you should specify the *.html* extension for automatic file association.

-host *hostnames*

(Optional) Specifies the names of one or more hosts you want to report on.

If you do not specify a host, sereport uses *localhost*.

-r | report *number*

Specifies the report number to create.

seretrust Utility—Generate Commands to Retrust Programs and Secure Files

The seretrust utility generates the selang commands required to retrust programs and secure files defined in the database. The seretrust utility reports the status of the SECFILE and PROGRAM resources that are defined as trusted but have changed. seretrust also checks whether programs have been changed but have not yet been caught by the Watchdog. (This means that in the CA Access Control database, these programs are still marked as trusted.) These programs are added to seretrust output with a note that the program content or timestamp has been changed, and the program needs to be retrusted.

Note: On UNIX, programs with setuid and setgid bits are stored in the database with their full descriptions, including their inode values. If you restore the system from backups, the programs occupy different inodes. CA Access Control detects the mismatch between the inodes and marks all the trusted programs as untrusted. The seretrust utility locates the trusted programs that are defined in the database and updates their inode values, so that when you invoke CA Access Control, the trusted programs remain trusted.

If you do not specify any switches, only untrusted programs and untrusted secured files are processed.

This command has the following format:

`seretrust [-a] [-l|-m|-p|-s]` *path*

-a

Processes all trusted and untrusted objects.

-h

Displays the help for this utility.

-l

Extracts information about the programs and files from the database in the current directory.

If you omit this option, seretrust processes the database that CA Access Control uses.

-m

Calculates the signatures for all kernel modules. If the signature property of a kernel module record is not valid, seretrust updates it with the correct signature, which ensures that the kernel module is trusted. Signatures are used only for Linux kernel modules.

-p

Processes records in the PROGRAM class only.

-s

Processes records in the SECFILE class only.

path

Specifies the base path for searching programs and secure files that need to be retrusted.

The utility processes the specified directory and all subdirectories.

Example: Retrust untrusted programs and secure files

This example shows you how you can use the seretrust utility to retrust programs and secure files.

Note: This example shows you a sample command output on UNIX, but the utility works the same on Windows.

To retrust programs and secure files, follow these steps:

1. As the CA Access Control database administrator, enter the following seretrust command:

   ```
   seretrust > retrust_script
   ```

   The utility processes both trusted programs and secured files because you did not specify any options; it also uses the root path because you did not specify any base path.

   seretrust displays the following information on the screen:

   ```
   Retrusting PROGRAMs & SPECFILEs, Base path = /
   Total of 0 entries retrusted. (Class=SECFILE)
   Total of 16 entities retrusted. (class=PROGRAM)
   ```

   The following is the content of a script file seretrust can create:

   ```
   chres PROGRAM ("/usr/bin/chgrpmem") trust
   chres PROGRAM ("/usr/bin/chie") trust
   chres PROGRAM ("/usr/bin/crontab") trust
   chres PROGRAM ("/usr/bin/cu") trust
   chres PROGRAM ("/usr/bin/ecs") trust
   chres PROGRAM ("/usr/bin/newgrp") trust
   chres PROGRAM ("/usr/bin/rmquedev") trust
   chres PROGRAM ("/usr/bin/rsh") trust
   chres PROGRAM ("/usr/bin/sysck") trust
   chres PROGRAM ("/usr/bin/uuname") trust
   chres PROGRAM ("/usr/lib/methods/showled") trust
   chres PROGRAM ("/usr/lib/mh/post") trust
   chres PROGRAM ("/usr/lib/mh/slocal") trust
   chres PROGRAM ("/usr/lpp/X11/bin/xlock") trust
   chres PROGRAM ("/usr/lpp/X11/bin/xterm") trust
   chres PROGRAM ("/usr/sbin/chvirprt") trust
   ```

2. Run the selang script file seretrust created to retrust the programs and files:

   ```
   selang -f retrust_script
   ```

serevu Utility—Handle Unsuccessful Login Attempts

Valid on UNIX

The serevu utility handles users who have had a specified number of failed login attempts during a specified period. Depending on your specifications, it can disable, report, or ignore the user. By default, it disables the user in the UNIX environment of the local station. If no such user exists locally, serevu checks the NIS information to find the user.

If you set a value in the passwd_pmd configuration setting, CA Access Control updates the appropriate PMDB, which then propagates the update to its subscribers. If you did not set a value in the passwd_pmd token, CA Access Control uses the value in the parent_pmd configuration setting, which then propagates the update to its subscribers.

Note: If you want serevu to send commands to the PMD (which, you can configure in serevu.cfg) and root is not defined on the PMD with the ADMIN attribute or with terminal access, you should define the following on the PMD and all of its subscribers:

```
eu _serevu logical
authorize admin USER uid(_serevu) access(a)
# The following line can be executed on the master PMD only
authorize terminal localTerminalName uid(_serevu) access(a)
```

Notes: For the serevu utility to work properly, the user root must have write access to the file /etc/passwd. If you define a remote computer in the serevu configuration file (serevu.cfg), you must also give login authorization to the remote computer. For example:

```
eu _serevu admin logical
authorize terminal localTerminalName uid(_serevu) access(a)
er specialpgm $ACDIR/bin/serevu seosuid(_serevu ) unixuid(root)
```

This command has the following format:

```
serevu {daemon|nodeamon} [-f nn] \
    [-d {nn[s|m|h|d|w]|FOREVER}] \
    [{-s|-t} nn[s|m|h|d|w]]
```

daemon

Runs the utility as a daemon. This is the default value.

nodaemon

Runs the utility as a regular process.

-d

Specifies the amount of time for which the user's login is disabled. By default, this value is in seconds.

Note: The amount of time a user account is disabled cannot be less than the amount of time between each serevu scan. The amount of time a user account is disabled should be a multiple of the time between each serevu scan.

-f

Specifies the number of failed logins. The serevu utility disables the accounts of users who reach this number of failed logins over the specified period.

Note: We recommend that the number of failed logins, which can also be defined by the value of the *def_fail_count* configuration setting, always be the same as the value of allowed unsuccessful login attempts set on your system. (On Solaris, for example, the system values for this are set in /etc/default/login by the RETRIES token.) See your operating system documentation for more details.

-h

Displays the help for this utility.

-s

Specifies the time period, starting from *now* and going backwards, within which serevu scans for failed logins.

Default: 300 seconds (configuration setting).

-t

Specifies the time period that should elapse between successive serevu checks.

Default: 120 seconds (configuration setting).

FOREVER

Used with the -d option, specifies the time as unlimited. If you use this parameter, user logins will be disabled forever.

***nn*[s|m|h|d|w]**

Used with the -d, -s, and -t options, specifies the time for the option.

s

nn in seconds (the default).

m

nn in minutes.

h

nn in hours.

d

nn in days.

w

nn in weeks.

sessfgate Utility—Route Unicenter Security Requests to CA Access Control

The sessfgate utility routes and reformats Unicenter Security APIs from the message queue to CA Access Control. The Unicenter Security APIs on UNIX all channel into a message queue. The sessfgate utility processes the API requests sent through the message queue and routes these reformatted and rerouted requests to CA Access Control. The utility then translates the return codes of CA Access Control to Unicenter TNG equivalents.

To activate the gateway, you must run the Unicenter Integration setup procedure. The Unicenter Integration setup installs the sessfgate program in the *ACInstallDir*/tng/bin directory (where *ACInstallDir* is the directory where you installed CA Access Control, by default /opt/CA/AccessControl/). After Unicenter Security is shut down and CA Access Control is started, sessfgate accepts API requests instead of SSF.

This command has the following format:

```
sessfgate [-i|-s|-l] -t
```

-I

Specifies to start the gateway.

-s

Specifies to stop the gateway.

-l

Specifies the status.

-t

Toggles the tracing file (log file = /opt/CA/AccessControl//log/sessftrace.log).

Note: If you run seload before running Unicenter TNG, you must start sessfagte manually with the following command:

ACInstallDir/tng/bin/sessfagte -I

where *ACInstallDir* is the directory in which you installed CA Access Control.

sesu Utility—Substitute User

The sesu utility lets you temporarily act as another user. This utility is the CA Access Control version of the UNIX su command. However, the sesu utility provides a user substitution command that does not require you to provide the password of the substituted user. The authorization process is based on the CA Access Control access rules as defined in class SURROGATE and, optionally, on the password of the user executing the command.

The sesu utility uses the tokens in the sesu section of the seos.ini file. It also uses the following special files:

- /etc/passwd
- /etc/group
- /etc/shells

To protect against inadvertent use, sesu is marked in the file system so that no one can run it. The security administrator must mark the program as executable and setuid to root before you can use it.

Important! Before you use the sesu utility, define all users to the CA Access Control database and set sesu prerequisites. This prevents you from opening up the entire system to users who are not defined to CA Access Control.

Usage notes:

- If the CA Access Control authorization server is not found, the utility executes the system's standard su command.
- If the sesu.old_sesu configuration token is set to no, the utility executes the system's standard su command.
- If /etc/shells exists, and it does not specify the current shell, sesu does not permit substitution to root.

This utility has the following format:

`sesu [-] [username] [-l] [-n] [-s shell] [-c command]`

-

Sets the environment to that of the target user.

Note: On Linux, this is the same as using the *-l* option.

-c *command*

Executes the specified command then exits.

Enclose commands containing spaces in quotes.

-h

Displays the help for this utility.

-l

(Linux only). Specifies that the shell it opens is a login shell.

-n

Specifies not to prompt the user for password

Important! When used, the utility runs as the root account and performs a LOGIN event.

Note: If the security authorization server is not found, the utility uses /bin/su.

-s *shell*

(Linux only). Specifies a shell to open instead of the shell from the user's passwd entry.

The shell must be listed in the /etc/shells file.

username

Changes the ID associated with the session to the ID of the specified target user *username*.

If you do not specify a *username*, sesu default to root.

Examples

- The following command changes the UID to root. The environment remains that of the user who executed the command.

  ```
  sesu
  ```

- The following command changes the UID to root. The utility changes the environment to root's environment.

  ```
  sesu -
  ```

- The following command surrogates to the user John.

  ```
  sesu John
  ```

- The following command surrogates to the user Carol and executes the specified command, ls -la, from the /home/carol directory.

  ```
  sesu - Carol -c "ls -la /home/carol"
  ```

- The following command surrogates to the user Angelo, uses a bash shell and opens it as a login shell.

  ```
  sesu Angelo -l -s /bin/bash
  ```

  Note: This is valid on Linux only.

sesudo Utility

The sesudo utility executes commands for one user with the permissions of another user. This lets regular users perform actions that require administrator authority.

The rules governing user authority to perform commands in this way are defined as access rules in the SUDO class. A record in the SUDO class contains a command script, and can specify both users who are permitted to run the script with sesudo and users who are forbidden to.

sesudo Utility—Execute a Command as Another User on UNIX

Valid on UNIX

The sesudo utility executes commands for one user with the permissions of another user. The sesudo utility borrows the permissions of another user (the *target* user) to perform one or more commands. This allows regular users to perform, for example, actions-such as the mount command-that require superuser authority.

The rules governing user authority to perform commands in this way are defined as access rules in the SUDO class. A record in the SUDO class contains a command script, and can specify both users who are permitted to run the script with sesudo and users who are forbidden to.

Each time sesudo runs, it returns one of the following values.

-2

Target user not found, or command interrupted

-1

Password error

0

Execution successful

10

Problem with usage of parameters

11

syscall is not loaded

20

Target user error

22

syscall is loaded but the daemon is not running

30

Authorization error

This command has the following format:

```
sesudo {-h|-list|record [params]}
```

-h

Displays the help screen.

-list

Lists sesudo commands you can execute. These are the SUDO records defined in the CA Access Control database that you are authorized to execute.

record

Specifies the name of the SUDO class record the security administrator gave to the command you want to execute using the sesudo utility.

params

(Optional) Specifies the parameters you want to send to the command you are executing.

sesudo Utility—Execute a Command as Another User on Windows

Valid on Windows

The sesudo utility executes commands for one user with the permissions of another user. The sesudo utility borrows the permissions of another user (the *target* user) to perform one or more commands. This allows regular users to perform, for example, actions-such as the mount command-that require superuser authority.

The rules governing user authority to perform commands in this way are defined as access rules in the SUDO class. A record in the SUDO class contains a command script, and can specify both users who are permitted to run the script with sesudo and forbidden users.

Note: The user executing the program invoked by sesudo cannot be changed from CA Access Control for Windows.

This command has the following format:

```
sesudo {-h|-list|-do record [params]}
```

-h

Displays the online help screen.

-list

Lists sesudo commands you can execute. These are the SUDO records defined in the CA Access Control database that you are authorized to execute.

-do *record* [*params*]

Specifies that sesudo executes a command as another user.

record

Specifies the name of the SUDO class record the security administrator gave to the command you want to execute using the sesudo utility.

params

(Optional) Specifies the parameters you want to send to the command you are executing.

seuidpgm Utility—Extract Trusted Programs

Valid on UNIX

The seuidpgm utility extracts all the programs whose Set-User-ID bit or Set-Group-ID bits are on. seuidpgm traverses a file system and creates the selang commands for adding these programs to the PROGRAM class.

seuidpgm creates the commands in the selang command language and writes them to the standard output. You can use a pipeline to the selang utility, or redirect the output to a file. We recommended that you redirect the output to a file, because then you can edit the output to remove unwanted programs or add additional programs. Use this procedure to search for undesirable setuid programs in your system.

Note: We recommended that you run the UxImport utility to define users and groups before running the seuidpgm utility. However, if you have not run UxImport, you can use seuidpgm with the -g and -u options to define users and groups.

seuidpgm descends through the paths specified at the command line to all subdirectories of the starting path. Multiple start paths are allowed.

You can specify any number of options. When specifying more than one option, separate the options with spaces.

If a program is a setuid program and has write access, seuidpgm treats the program like all other setuid programs, but also sends a warning to standard error.

Note: For more information on how to control PROGRAM class records, see the *Endpoint Administration Guide for UNIX*.

This command has the following format:

```
seuidpgm option startDir ... [-x excludeDir]
```

-d

Automatically creates entries for setuid and setgid programs in the PROGRAM class, with defaccess set to execute, instead of analyzing the file permissions in UNIX to determine the permitted file access. In some cases, one setuid or setgid program executes another one. If you do not include this option, the program trying to execute the setuid or setgid program is *not* able to execute it.

We recommend that you use this option.

-f

Creates rules for both the FILE and PROGRAM classes.

-g

Creates GROUP records for setgid programs.

Note: Use this option *only* if you have *not* run UxImport.

-l

Creates a single permit for programs which have *hard* or *symbolic* links.

If you want to scan your file system from some directories only (not from the root directory) and to include the -l option, use multiple starting paths on the command line; otherwise the -l option may be inefficient.

-n

Does not traverse NFS at all.

We recommend that you use this option.

-o

Writes the file names to the standard output but does not create selang commands.

-p

Enables setuid programs from NFS directories, but only when the mount table allows setuid from that mounted file system.

-q

Runs the utility in Quiet-Mode; error messages are not sent to standard error.

-s

Creates entries for setuid/setgid programs in class SECFILE, instead of creating entries for the PROGRAM class.

-u

Creates USER records for setuid programs.

Note: Use this option *only* if you have *not* run UxImport.

-x *excludeDir*

Excludes a directory from the tree. The specified directory is not searched for setuid and setgid programs. This option must be the last option specified in the command line. *Path* is the full path of the directory to be excluded. To exclude more than one directory, repeat the -x option for each directory.

startDir

Specifies a space-separated list of top directories to search for trusted programs.

Examples

- The following command prints selang commands to add all programs with set-user-id or set-group-id turned on, defaccess execute, checking for duplicate names or the same inode, in quiet mode, and without passing through NFS. The program scans from the /usr directory and its subdirectories, the /var directory and its subdirectories, and the /etc directory and its subdirectories. Output is directed to the file seprogs.seos in your home directory.

  ```
  seuidpgm -dlqn /usr /var /etc > ~/seprogs.seos
  ```

  The output should look similar to the following:

  ```
  ## ************************************************
  ## seuidpgm List Sun Feb 9 14:24:16 1997
  # Start Path= /usr
  # ************************************************
  nr PROGRAM /usr/lpp/bos/inst_root/lpp/inu_LOCK defaccess(EXEC)
  nr PROGRAM /usr/lpp/X11/bin/xlock defaccess(EXEC)
  nr PROGRAM /usr/bin/setsenv defaccess(EXEC)
  nr PROGRAM /usr/bin/shell defaccess(EXEC)
  nr PROGRAM /usr/bin/su defaccess(EXEC)
  nr PROGRAM /usr/bin/sysck defaccess(EXEC)
  nr PROGRAM /usr/bin/tcbck defaccess(EXEC)
  nr PROGRAM /usr/bin/usrck defaccess(EXEC)
  nr PROGRAM /usr/bin/vmstat defaccess(EXEC)
  ```

- The following command scans the root directory and all its subdirectories, except the /home directory:

  ```
  seuidpgm -qln / -x /home
  ```

More information:

UxImport Utility—Extract Information from the UNIX Operating System (see page 246)
selang Utility—Run the CA Access Control Command Line (see page 165)
seoswd Daemon (see page 273)
seosd Daemon (see page 268)

seversion Utility—Display CA Access Control Program Module Version Information

Valid on UNIX

The seversion utility displays information regarding the version of a CA Access Control module. You can display the following data:

- The global and minor version numbers.
- The date and time that the module was compiled.
- The station that the module was compiled on.

This command has the following format:

```
seversion [-a|-l|-g|-h|-m|-s|-5] module
```

-a

Displays the requested information in table format.

-g

Displays only the global version number, omitting titles.

-h

Displays the help for this utility.

-l

Displays included library information.

-m

Displays only the minor version number, omitting titles.

-s

Displays SHA1 signature, omitting titles.

-5

Displays the MD5 signature, omitting titles.

This option works only while not in FIPS-only mode.

module

Specifies the file name of the module whose version number you want to display.

Example

To display version information for the sesudo utility, enter the following command:

```
seversion /opt/CA/AccessControl//bin/seosd
```

A message similar to the following appears on the screen while not in FIPS mode:

```
CA Access Control seversion vX.X.X.xxx - Display module's version
Copyright (c) YYYY CA. All rights reserved.
Running under:  Linux
File name: /opt/CA/AccessControl//bin/seosd
Version  : major.minor.sp.build
Created  : MMM DD YYYY hh:mm:ss
OS info  : i86PC
SHA1     : 10068CC6A70195B84AF896682CCBA1A4B7B43CD1
MD5:     : 1F9BD56CA523A33FFBC47551ECE093E5
```

sewhoami Utility—Display Your CA Access Control User name and Security Credentials on UNIX

Valid on UNIX

The sewhoami utility displays the user name as it is known to the CA Access Control authorization daemon. sewhoami is similar to the whoami utility provided by UNIX systems, but it produces different and often more useful information:

- If the user executes an su command and then executes the UNIX whoami utility, it displays the user name according to the user ID acquired after executing the su command.
- If the user executes an su command and then executes the CA Access Control sewhoami utility, it displays the original login ID of the user; it also displays authorization information.

This command has the following format:

```
sewhoami [-a|-d]
```

-a

Displays the user's credentials; that is, the contents of the user's ACEE.

Note: For more information on the ACEE, see the *Endpoint Administration Guide for UNIX*.

-d

Displays the ACEE handle associated with the user and the handle's name in the database.

Example: Display Your CA Access Control User Name and Security Credentials on UNIX

This example displays your own user name and security credentials as they are known to the CA Access Control authorization daemon:

```
sewhoami -a
```

If you are a root user, the sewhoami output may look like the following example:

```
root
ACEE Contents
  User's Name             : root
  ACEE's Handle           : 52
  Group Connections Table:
    Group Name              Connection Mode
            ==================    ================================
    adm                     Regular
    bin                     Regular
    daemon                  Regular
    disk                    Regular
    root                    Regular
    seosaudt                Regular
    sys                     Regular
    wheel                   Regular
Categories              : <None>
Profile Group           : <None>
Security Label          : <None>
User's Audit Mode       : Failure LoginSuccess LoginFailure
User's Security Level   : 0
Source Terminal         : <Unknown>
Process Count for ACEE  : 19
User's Mode             : Admin Auditor
ACEE's Creation Time    : Tue Mar 17 14:53:07 2009
```

If you are a user named test, and are not a root user, the sewhoami output may look like the following example:

```
test
ACEE Contents
  User's Name             : test
  ACEE's Handle           : 65
  Group Connections Table:
    Group Name              Connection Mode
           ==================    ================================
    seosaudt                Regular
    users                   Regular
Categories              : <None>
Profile Group           : secadmin
Security Label          : <None>
User's Audit Mode       : Failure LoginSuccess LoginFailure
User's Security Level   : 0
Source Terminal         : localhost.localdomain
Process Count for ACEE  : 2
User's Mode             : Admin Auditor
ACEE's Creation Time    : Wed Mar 18 15:34:53 2009
```

More information:

secons -whoami Function—Display Your User Name and Security Credentials (see page 154)

uninstall_AC Utility—Remove CA Access Control from the Current Computer

Valid on UNIX

The uninstall_AC utility removes all or part of CA Access Control from the station on which you execute the command. The default (-all) removes the entire product from the station.

Note: The CA Access Control kernel extension should be unloaded prior to uninstall.

This command has the following syntax:

`uninstall_AC [-all | -admin] [-f] [-force] [-h] [-ignore_dep] [-d path] [-fn file]`

-admin

Removes only administration tools such as Security Administrator and seauditx from the station.

Note: The *admin* package is no longer included with CA Access Control. This option is used for removing older versions of CA Access Control.

-all

Removes the entire product from the station.

-d *path*

Defines the directory where CA Access Control is installed.

Note: If CA Access Control is installed in the default directory (/opt/CA/AccessControl/) you do not need to specify this option.

-f

Removes CA Access Control in silent mode.

-fn *file*

Executes the specified file after the uninstall completes.

-force

Forces uninstall to proceed even if the kernel extension unload process fails.

-h

Displays the help for this utility.

-ignore_dep

Specifies that the uninstallation procedure will not check for dependency with other products.

Example: Completely remove CA Access Control from a computer

To completely remove CA Access Control from this computer, if it was installed in the default directory, enter the command:

```
uninstall_AC
```

uxauthd.sh Script—Administer UNIX Authentication Broker Agent

Use the uxauthd.sh script to administer the UNIX Authentication Broker agent. We recommend that you use the uxauthd.sh script to administer the UNIX Authentication Broker agent because this help ensures that the environment is configured correctly.

The uxauthd.sh script is located in the following directory, by default: /opt/CA/uxauthd/lbin.

This command has the following format:

```
uxauthd.sh {start | stop | restart | status | debug level}
```

start

Starts the UNIX Authentication Broker agent.

stop

Stops the UNIX Authentication Broker agent.

restart

Restarts the UNIX Authentication Broker agent

status

Displays the status of the UNIX Authentication Broker agent. The status states are:

- uxauthd running
- uxauthd not running

debug *level*

Specifies to start the UNIX Authentication Broker agent in debug level.

Range: 1-3

Note: Using uxauthd.sh to start or stop the UNIX Authentication Broker agent affects the status of the Report Agent.

uxauth_selinux.sh—Enable SElinux Support

The uxauth_selinux.sh script deploys a policy that enables UNIX Authentication Broker to work in SElinux environment. The script enables support for the following utilities: ssh, rlogin, ftp, sftp and passwd.

The uxauth_selinux.sh script is located in the following directory, by default: /opt/CA/uxauthd/lbin.

This command has the following format:

```
uxauth_selinux.sh {-i [-e]| -r | -h}
```

-i

Installs the policy in the SElinux environment

-e

Specifies to invoke the extensive installation option that adds permissions for the usr_t type

-r

Removes the policy from the SElinux environment

-h

Displays the help

uxconsole Utility—Manage UNIX Authentication Broker Endpoints

The uxconsole utility lets you manage your UNIX Authentication Broker endpoints. You use the uxconsole utility to display information about the UNIX Authentication Broker installation, register the UNIX Authentication Broker endpoint in Active Directory, and manage and migrate users and groups.

The utility handles several tasks and has the following functions:

Task	Function
Register UNIX computers in Active Directory	uxconsole -register (see page 233)
Deregister UNIX computers in Active Directory	uxconsole -deregister (see page 233)
Set verbosity level	uxconsole -debug
Activate login for Active Directory users	uxconsole -activate

Task	Function
Deactivate login for Active Directory users	uxconsole -deactivate
Manage users mapping	uxconsole -map (see page 227)
Migrate users and groups to Active Directory	uxconsole -migrate (see page 231)
Manage users and groups	uxconsole -manage (see page 229)
Display endpoint status	uxconsole -status (see page 236)
Perform Kerberos operations	uxconsole -krb (see page 239)
Perform LDAP queries in Active Directory	uxconsole -ldap (see page 240)
Display UNAB NSS cache data	uxconsole -dbdump (see page 242)
Verify Active Directory user accounts	uxconsole -verify (see page 244)

uxconsole -map—Manage Users Mapping

Valid on UNIX

Use the map command to NIS or local user accounts to Active Directory user accounts.

Note: When you use the -map option, the uxconsole utility does not connect to Active Directory to identity conflicts in user account details.

This command has the following formats:

```
uxconsole -map [-add] [-scope {l|n|a}] { -all | <unix_name> [ <ad_name>] | -input
<file> } [-d <domain>  ] [-force] [-v]
uxconsole -map -local { <unix_name> | -input <file> } [-force] [-v]
uxconsole -map -del   { <unix_name> | -input <file> } [-force] [-v]
uxconsole -map -show [<filter>] [-scope {l|n|a}] [-v]
uxconsole -map -h
```

-add

Add users mapping

-scope {l|n|a}

Specifies the mapping scope:

- l—map the local user accounts only
- n—map NIS/NIS+ user accounts only
- a—map local and NIS/NIS+ user accounts

-all

Specifies to map all NIS and or local user accounts with identical user names.

<unix name>

Specifies to map a single UNIX user account, either NIS or local account.

Note: You can also use this parameter to delete mapped NIS or local user accounts.

-ad <*ad name*>

Specifies the Active Directory name of the local user account.

-input <*file*>

Specifies an input file containing mapping requests. Create the map file in a CSV format with the following fields and parameters:

```
<unix_name>,[<ad_name>],[<domain>]
```

<unix_name>

Specifies the UNIX user account name.

[<ad_name>]

Specifies the Active Directory user account. *ad_name* is an optional parameter. If you do not specify the Active Directory user account, the account is mapped to an AD account with the same UNIX user account name.

[<domain>]

Specifies the domain name of the Active Directory account. *domain* is an optional parameter.

-d <*domain*>

Defines the Active Directory domain name that contains the user account.

Note: You can specify the full user credentials using the following format: *<name>@<domain>*. If the domain name is not specified, then the domain is mapped to the registration domain.

-force

Specifies to force user mapping and overwrite existing mapping or migration status or delete user mapping.

Note: By default, uxconsole does not delete partially migrated user accounts.

-local

Specifies to set the user account as a local exception.

Note: If you specify a user as local exception, UNIX Authentication Broker does not manage the user account, although an identical user account may exist in the Active Directory.

-del

Specifies to delete local or NIS user mapping.

-show

Specifies to display users mapping details.

<filter>

Defines the wildcard that returns a subset of users.

-v

Specifies to activate verbosity.

-h

Displays the help.

uxconsole -manage—Manage Users and Groups

Valid on UNIX

Use this command to list or information for local or enterprise users and groups.

This command has the following formats:

```
uxconsole -manage {-find | -show [-detail]} {-user <filter> | -group <filter>}
uxconsole -manage -show -policy
```

-find

Specifies to display a list of local and enterprise users or groups.

-show

Specifies to show the details of a specific user or group, a subset of users and groups, or to show policies.

-detail

Specifies to display the user settings in detail.

-user *filter*

Defines the wildcard that returns a subset of users.

-group *filter*

Defines the wildcard that returns a subset of groups.

-policy

Specifies to display the enterprise login policy.

Example: Display User Status

The following example shows you the output for a local UNIX user (local1) who is mapped to an Active Directory user with a different name (ent1). The Active Directory user has UNIX attributes enabled, so can log in to the UNIX Authentication Broker endpoint:

```
uxconsole> ./uxconsole -manage -show -detail -user ent1
CA Access Control UNAB uxconsole v12.52.0.160 - console utility
Copyright (c) 2009 CA. All rights reserved.

USER 'ent1' information
-----------------------------------------------------
Type                : Local User
Login Name          : local1
Mapped to           : ent1@example.com
Enterprise Account  : Enabled
Local Account       : Enabled
Login               : Allowed
Login Reason        : User exists locally
Uid                 : 300
Gid                 : 101
Shell               : /bin/bash
Home Directory      : /home/local1
Unix Groups           : 30017(unabca_gx2), 30016(unabca_gx1)
All Groups            : unabca_gw1@company.com, unabca_gw2@company.com,
unabca_gx1@company.com

Type                : Enterprise User
Login Name          : ent1
Principal Name      : ent1@example.com
Enterprise Account  : Enabled
Login               : Allowed
Login Reason        : According to internal default
Uid                 : 10133
Gid                 : 13870
Shell               : /bin/sh
Home Directory      : /home/ent1
```

uxconsole -migrate—Migrate UNIX Users and Groups to Active Directory

Valid on UNIX

Using the migrate command migrates users and groups from the UNIX host into Active Directory. The migration process attempts to migrate local users and groups into Active Directory and disable the local accounts.

This command has the following format:

```
uxconsole -migrate [-scope {l|n|a}] {-mode {p|f}|-input file} [-emulate] [-d domain]
[-a name [-w pass]] [-users] [-groups] [-cgc container] [-new] [-v level] [-h]

uxconsole -migrate [-show {-user filter|-group filter}]
```

-migrate

Defines the UNIX users migration option.

-scope {l | n | a}

Specifies the migration scope:

- l—migrate only local users and groups.
- n—migrate NIS users and groups from NIS\NIS+ server.
- a—migrate local and NIS/NIS+ users and groups.

Default: l

-mode {p | f}

Specifies the migration mode.

Options: partial, full

Default: f

-input *file*

Defines the full path of the accounts map file.

Note: Use the mapping file to resolve conflicts in user accounts that were discovered during the migration process. Create the map file in a CSV format with the following fields and parameters:
`type <USER|GROUP>, UNIX name <username>,requested action <KEEPLOCAL|MIGRATE|MAP>, AD name <AD mapped name>`

Example: USER,uxuser, MAP,aduser.

Important! You cannot specify the GROUP type to use the MAP action.You can use the MAP option to map user accounts only.

-emulate

Specifies that the migration process runs in emulation mode.

Note: Running the uxconsole -migrate command in emulation mode does not migrate users to Active Directory. In emulation mode the uxconsole creates a journal file that reports on possible conflicts in users and groups IDs. Use the emulation mode to resolve conflicts between UNIX and Active Directory users and groups IDs.

-d *domain*

Defines the name of the domain to migrate users and groups to.

Note: Running the -migrate -d command without supplying the administrator credentials does not enable UNIX Authentication Broker to migrate users and groups to Active Directory.

-a *name*

Specifies the Active Directory administrator used to register, create, and update users properties in Active Directory.

Note: Running the -migrate command without supplying the administrator credentials does not enable UNIX Authentication Broker to append UNIX attributes nor to add accounts or groups to Active Directory. You cannot resolve conflicts that were discovered during migration without supplying the Active Directory administrator credentials.

-w *passwd*

Specifies the Active Directory administrator's account password.

-users

(Optional) Specifies that only users are migrated to Active Directory.

Note: If not specified, all the users are migrated to Active Directory.

-groups

(Optional) Specifies that only groups are migrated to Active Directory.

Note: If not specified, all the groups are migrated to Active Directory.

-cgc *container*

Specifies the name of the Active Directory container where new groups are created.

-new

Specifies to migrate only new users and groups that were not not previously migrated.

-v *level*

Specifies the verbose level.

Range: 1-5

-h

Displays the help.

-show

Displays users and groups migration information.

Note: If specified, users and groups are not migrated.

-user *filter*

Displays only those users that match the filter criteria.

-group *filter*

Displays only those groups that match the filter criteria.

uxconsole -register—Register UNIX Computers in Active Directory

Valid on UNIX

Use this command to register the UNIX host in Active Directory. Registering the UNIX host is part of the UNIX Authentication Broker configuration process that lets Active Directory users log in to the UNIX host.

Note: After you register the UNIX host, you must activate UNIX Authentication Broker to let Active Directory users log in to the host.

The utility cannot register the UNIX host in the following circumstances:

- If the host name of the UNIX computer without the domain suffix contains more than 15 characters, registration fails. This is because Active Directory imposes NetBIOS-based restrictions on the number of characters in the name of computer objects.

  For example, you cannot register a UNIX computer named engineering-dept-sol2 in Active Directory because the host name contains more than 15 characters. You can register a UNIX computer named eng-dept-sol2.example.com because the host name without the domain name (eng-dept-sol2) contains less than 15 characters. To display the host name of the UNIX computer, run the hostname command.

- If all DCs in the Active Directory site that the UNIX host uses to communicate with Active Directory are specified in the ignore_dc_list configuration setting in the ad section of the uxauth.ini file, registration fails.

  When you register a UNIX host in Active Directory, by default, the uxconsole utility automatically discovers the Active Directory site that is closest to the physical location of the endpoint, and communicates only with DCs in this site. You can also use the -t option to specify this Active Directory site.

You can run this command multiple times on the same computer. For example, you can run this command to repair the UNIX Authentication Broker host registration with Active Directory if the keytab file is deleted.

Note: You can run the uxconsole - register command without arguments to use the default settings. The program prompts you for additional information required.

This command has the following format:

```
uxconsole -register [-a name] [-w pass] [-d domain] [-v level] [-n] [-o container] [-s server] [port #] [-h] [-t site] [-sso]
```

```
uxconsole -deregister [-a name] [-w pass] [-v level] [-o container] [-s server] [port #]
```

-register

Specifies that Active Directory registers UNIX Authentication Broker.

-deregister

Specifies that Active Directory deregisters UNIX Authentication Broker.

-a *name*

Defines the name of a user that has privileges for registering computers in Active Directory.

Default: administrator

-w *pass*

Defines the password of the user that has privileges to register computers in Active Directory.

-d *domain*

Defines the domain name the Active Directory is part of.

-h

Displays the program help.

-n

Specifies that the uxauthd agent runs after the registration process completes.

If you do not specify this option, uxauthd does not run after the registration process completes.

-o *container*

Defines the Active Directory container name where the UNIX computer is registered.

Note: The Active Directory container must exist before you register the UNIX computer.

-port *#*

Defines the Active Directory listening port number.

-s *server*

Defines the Active Directory server name.

-sso

Specifies that the uxconsole manages Kerberos files for Single Sign On (SSO)

-t *site*

Defines the Active Directory site that contains the DCs that UNIX Authentication Broker uses to communicate with Active Directory, and writes the name of the site to the ad_site configuration setting in the ad section of the uxauth.ini file.

We recommend that you do not specify this option. If you do not specify this option, the utility automatically selects the best Active Directory site to use.

Note: The values in the ignore_dc_list and lookup_dc_list configuration settings affect how UNIX Authentication Broker implements Active Directory site support.

-v *level*

Defines the verbose level to use during the installation process.

Example: Register a UNIX Host in Active Directory

This example shows you how to register a UNIX computer in Active Directory. You type in the user name (-a administrator) and password (-w admin), set the verbosity level (-v 3), specify that the UNIX Authentication Broker agent does not run at the end of the installation (-n), and define the name of the container in Active Directory (-o OU=COMPUTERS). The container must exist before you register the UNIX computer in Active Directory:

```
./uxconsole -register -a administrator -w admin –v 3 -n -o OU=COMPUTERS
```

uxconsole -status—Display UNIX Authentication Broker Status

Valid on UNIX

Use this command to display the the status of UNIX Authentication Broker on the endpoint. Using the -detail argument displays all the available information about the status of UNIX Authentication Broker.

This command has the following format:

```
uxconsole -status [-detail]
```

-status

Specifies to display the UNIX Authentication Broker status.

-detail

Specifies to display the UNIX Authentication Broker status in detail.

Example: Display the UNIX Authentication Broker Status in Detail

The following example shows you the output you receive when you run the uxconsole -status -detail command.

```
#./uxconsole -status -detail
CA Access Control uxconsole v12.52.0.160 - console utility
Copyright (c) 2009 CA. All rights reserved.

Registration domain    - example.com
DCs                    - computer1, computer2
User search base       - DC=unixauth,DC=example,DC=com
User search filters
              Include   - CN=Users; OU=Test
             Exclude   - OU=WrongOU
Group search base      - CN=Users,DC=example,DC=com
Group search filters
             Exclude   - OU=Computers
Trusted domain         - DC=unab,DC=example,DC=com
DCs                    - winserver
User search base       - DC=unabdom,dc=example,dc=com
User search filters
            Include    - CN=users
Group search base      - DC=unab,DC=example,DC=com
UNAB mode              - full integration
UNAB status            - activated
Agent status           - running, pid = 6178
Time sync              - enabled (NTP server: 192.168.1.100)
Enterprise policy      - login@computer.com (updated: Mon Oct 19 14:36:47 2009)
Enterprise policy      - loginHG@GHNODE#01 (updated: Mon Oct 19 14:36:47 2009)
Local policy           - enabled
Default login access   - deny
AD Unix users          - 16 (updated: Sun Oct 19 15:53:04 2009)
AD Unix groups         - 8 (updated: Sun Oct 19 15:53:04 2009)
AD Windows groups      - 19 (updated: Sun Oct 19 15:53:04 2009)
Migration              - not migrated
CA Access Control      - installed
                         Include AD users and groups in AC ladb : yes
                         Display AD names in AC Audit : no
                         Support AD non-Unix groups in AC: yes
                         PAM authentication in AC utilities : yes
```

In this example, the output displays the following information:

- The Active Directory domain name—example.com
- The DCs with which the endpoint communicates—computer1, computer2
- The user and group search base filters
- The trusted domain—unab.example.com

- UNAB mode—full integration
- UNAB status—activated
- UNAB agent (uxauthd) status—running, pid = 6178
- Whether time synchronization was activated—enabled
- The NTP server IP address—192.168.1.100
- The name of deployed enterprise login polices—login@computer.com, loginHG@GHNODE#01
- When the enterprise login policies were last updated—updated: Mon Oct 19 14:36:47 2009
- Whether local login policy is activated—enabled
- Whether the default login policy is enabled—deny
- The number of UNIX users in Active Directory—16 and the time that they were last updated
- The number of UNIX groups in Active Directory—8 and the time that they were last updated
- The number of Windows groups in Active Directory—19
- The time that the UNIX users and groups and Windows groups were last updated—updated: Sun Oct 19 15:53:04 2009
- The migration status of the users—not migrated
- Whether CA Access Control is installed on this endpoint—installed
- Whether to include information regarding Active Directory users and groups in the CA Access Control ladb—yes
- Whether to display Active Directory users and groups names in CA Access Control audit records—yes
- Whether CA Access Control supports non-UNIX Active Directory groups—yes
- Whether to support PAM authentication in CA Access Control utilities—yes

uxconsole -krb—Perfrom Kerberos Operations

Valid on UNIX

Use this command to perfrom Kerberos operations from the UNIX Authentication Broker endpoint, for example, creating tickets. You do not need to install Kerberos on the endpoint to perform Kerberos operations.

This command has the following format:

```
uxconsole -krb [-init | -list | -passwd | -vno | -destroy |-resolve
```

-init

Specifies to obtain and cache a ticket

-list

Displays the content of a credentials cache or keytab

-passwd

Specifies to change the user Active Directory password

-vno

Displays the key version number for Kerberos principals

-destroy

Specifies to destroy the credentials cache

-resolve

Specifies to resolve a host name or IP address

Example: Obtain a Ticket Granting Ticket (TGT) using UNIX Authentication Broker keytab

The following example shows how you obtain a TGT using UNIX Authentication Broker keytab:

```
./uxconsole -krb -init -k
```

Example: List the content of the credentials cache

The following example shows how you list the content of the credentials cache:

```
./uxconsole -krb -list
```

Example: List the content of the keytab with encryption data

The following example shows how to display the content of the keytab including available encryption information:

```
./uxconsole -krb -list -keytab
```

uxconsole -ldap—Perfrom LDAP queries in Active Directory

Valid on UNIX

Use this command to perform LDAP queries on Active Directory from a UNIX Authentication Broker endpoint that does not have LDAP installed. Use this command instead of the ldapsearch utility. You can use this command to troublehsoot UNIX Authentication Broker installation, For exmpale, you can query Active Directory for the container to use.

Important! Verify that you have a Ticket Granting Ticket (TGT) before you use this command. You can obtain a TGT using the command: uxconsole -krb.

Note: The LDAP filter must comply with "RFC 2254:

This command has the followinf format:

`uxconsole -ldap -search [-d DC] [-p port] [-b base] [-s scope] [filter [attributes]]`

-search

Specifies the search option

-d *DC*

Specifies the Domain Controller to query

-p *port*

Specifies the LDAP port to use

-b *base*

Specifies the search base

-s scope

Specifies the search scope

Default: sub

filter *[attributes]*

Specifies the filter and attributes to use

Note: If you do not specify a filter, the '(objectClass=*) is used. If you do not specify any attributes, the select all option ('*') is used.

Example: Display a DSE

The following examples shows how you display a DSE:

```
./uxconsole -ldap -search '(&(objectClass=user) (objectCategory=user) )'
```

uxconsole -dbdump—Display UNAB NSS cache data

Valid on UNIX

Use this command to display users and groups information from the UNIX Authentication Broker NSS database. You can use this command to view information about users and groups that are defined in Active Directory.

This command has the following format:

```
uxconsole -dbdump [table [item]]
```

table [*item*]

Specifies to display the content of the table and items.

Note: If you do not specify the table name, this command displays all available tables.

Example: Display all Active Directory users stored in cache

The following example shows how to display all Active Directory users stored in the endpoint cache:

```
./uxconsole -dbdump pw
```

Example: Display all Active Directory groups stored in cache

The following example shows how to display all Active Directory groups stored in the endpoint cache:

```
./uxconsole -dbdump -gr
```

uxconsole -debug—Set Verbosity Level for Modules

Valid on UNIX

Use this command to set the verbosity level per module. UNIX Authentication Broker also sends PAM and NSS debug information to debug information to files.

This command has the following format:

`uxconsole -debug -m mod [-v level]`

-m *mod*

Specifies the module to set the verbosity level

Options: nss, pam, agent, all

-v *level*

Specifies the verbosity level.

Limits: 0-5

UNIX Authentication Broker writes the debug information to the following files:

UNABInstallDir/log/debug/pam_debug
UNABInstallDir/log/debug/pam_debug.back
UNABInstallDir/log/debug/nss_debug
UNABInstallDir/log/debug/nss_debug.back

Note: If you set the verbosity level to more than 0 while the agent is not running, you receive a message indicating that the UNIX Authentication Broker PAM module was activated. UNIX Authentication Broker sends the debug information to the syslog only.

uxconsole -verify—Verify Active Directory User Account UNIX Attributes

Valid on UNIX

Use this command to verify that an Active Directory user account is ready for use by UNIX Authentication Broker. This command locates the user account and verifies that the UNIX attributes (login shell, home directory, UID and GID) are consistent with the values as they exist in the UNIX Authentication Broker user cache database.

Note: This command does not verifies the user password.

This command has the following format:

```
uxconsole -verify -user <user_name>[<user_name1>][<user_name2>...]
```

-user

Specifies to verify the user account UNIX attributes in Active Directory

<user_name>

Specifies the Active Directory user account.

Example: Verify Active Directory user account UNIX attributes

The following example shows how to verify Active Directory user account UNIX attributes:

```
./uxconsole -verify -user Joe
```

In this example, you use the -verify command to verify the user account Joe UNIX attributes. UNIX Authentication Broker does the following:

- Checks the /etc/shells file to verify that the login shell specifies is supports
- Verifies that the user name length consists with the limitations as imposed by the operating system
- Verifies that the home directory is specified
- Verifies that the UID is specified
- Verifies that the GID in specified

How uxconsole Discovers an Active Directory Site

When you register a UNIX Authentication Broker endpoint with Active Directory, by default the uxconsole utility discovers the closest Active Directory site and communicates only with domain controllers (DCs) in this site.

The following process describes how uxconsole discovers the closest Active Directory site:

1. The UNIX Authentication Broker endpoint queries the DNS for SRV (service) records in the following format:

   _ldap._tcp.dc._msdcs.*domainName*

   The DNS returns the records for DCs in the domain.

2. The endpoint accesses Active Directory by binding and authenticating to a DC returned in the previous query.

   Note: The endpoint can bind to any of the returned DCs.

3. The endpoint uses an LDAP query to search Active Directory for the site in which the endpoint resides. The query uses the following filters:

   - Base Dn—no value
   - Scope—Base
   - Attribute—Netlogon
   - DnsDomain—Fully-qualified domain name
   - ntver—6.00

   For example, Filter on (&(DnsDomain=example.company.com)(ntver=6.00))

   The DC returns the name of the site in which the endpoint resides.

   Note: The DC uses the endpoint IP address to determine the site in which the endpoint resides.

4. The endpoint queries the DNS for SRV records in the following format:

   _ldap._tcp.*LocalSiteName*._sites.dc._msdcs.*domainName*.

   The DNS returns the records for DCs in the site in which the endpoint resides. The endpoint communicates only with DCs in this site.

UxImport Utility—Extract Information from the UNIX Operating System

Valid on UNIX

The uximport utility extracts information from the UNIX operating system about the defined users, groups, terminals, hosts, and TCP services. It extracts information from NIS, if it is installed, and the system is configured accordingly. It also provides DNS support. You should use uximport as part of the installation procedure.

uximport automatically processed the extracted information to generate selang commands that you can use to add users and groups to the CA Access Control database. The generated commands are printed to the standard output. Use redirection to a file, or pipeline to the selang utility.

This command has the following format:

`UxImport` *switches* [*options*]

-a

Generates the selang commands required to import users, groups, and hosts, and to join users to their default groups.

-c

Generates the selang commands required to explicitly join users to their default groups.

Note: If you also import groups with the -g switch, CA Access Control generates the commands that join users to the groups to which they are explicitly linked.

-g

Generates the selang commands required to import groups from UNIX and NIS to the CA Access Control database.

-h

Generates the selang commands required to import hosts from UNIX, NIS, and DNS to the CA Access Control database. uximport extracts host information from the file /etc/hosts and from NIS, and builds HOST resources. For each host entry in the file /etc/hosts or extracted from NIS, the appropriate newres command is built, and permission to receive any TCP service is assigned to that host.

In addition, DNS is supported with the -d option. In some machines, information from the file /etc/hosts and NIS is ignored if the specified DNS daemon is running. In Solaris, the information gathered depends on the configuration of the system in the file /etc/nsswitch.conf.

-t

Generates the selang commands required to import terminal rules from UNIX and NIS to the CA Access Control database.

uximport extracts host information from the file /etc/hosts and from NIS, and builds TERMINAL resources. For each entry in the file */etc/hosts* or extracted from NIS, the appropriate newres TERMINAL command is built and permission to log in from the terminal is granted.

In addition, DNS is supported with the -d option. In some machines information from the file /etc/hosts and NIS is ignored if the specified DNS daemon is running. In Solaris, the information gathered depends on the configuration of the system in the file /etc/nsswitch.conf.

-T

Generates the selang commands required to import TCP services from UNIX and NIS to the CA Access Control database. The names are set according to GECOS in UNIX. The names are truncated to 40 characters if they are longer.

-u

Generates the selang commands required to import users from UNIX and NIS to the CA Access Control database. The actual user names are set according to GECOS in UNIX. The names are truncated to 40 characters if they are longer.

options

-d

Specifies the use of DNS for generating the list of hosts and terminals to import. Must be accompanied by the -h or -t switch.

-f

Skips search for multiple occurrences of the same name. By not using the standard uximport processes, this option handles the importing of many users and groups speedily, and saves memory. The -f option does not apply to hosts; you should combine them with one or more of the following switches: -u, -g, or -a. Also, use one of these switches when including the -c switch in conjunction with the -f option.

Join and surrogate rules are printed along with create records.

-G

Creates SURROGATE class rules for groups. uximport adds a record to the SURROGATE class for each group it defines, therefore making SURROGATE requests protected resources. It also adds rules so that root can surrogate to each of the groups.

-gr *n*

Specifies the number of grace logins for all users, forcing users to change their passwords after *n* logins. This ensures that the PASSWD_L_C property in the USER record is updated.

-o *owner*

Sets ownership rules for each record. We recommended that you use this option to prevent root from automatically becoming the owner of all the records. *Owner* is the name of the user or group to be assigned ownership of all records defined by uximport.

Note: You must specify this option as a separate argument followed by *owner*.

-pr *groupname*

Assigns a profile group to users. If you specify this option, CA Access Control uses that group when building a user's profile; otherwise, it uses the primary UNIX group.

-r

Specifies to continue scanning after a failure.

-s

Creates SURROGATE class rules for users and groups. The uximport function adds a SURROGATE record for every group it defines, thereby making SURROGATE requests to the group into protected resources.

-U

Creates SURROGATE class rules for users. uximport adds a record to the SURROGATE class for each user it defines, therefore making SURROGATE requests into protected resources. It also adds rules so that root can surrogate to each of the users.

-v

Displays the status of the program (verbose mode). We recommended that you use this option if your site has many users, groups, or hosts, so that you can verify the program's progress.

Example

The following command extracts all information of users, groups, and hosts from the UNIX and NIS databases. It then creates the selang commands that add those records to the database. uximport then creates SURROGATE class records and provides progress indication. Output is directed to the file uxinfo.seos in your home directory.

```
UxImport -a -s -v > ~/uxinfo.seos
```

More information:

seerrlog Utility—Display Error Log Records (see page 157)
selang Utility—Run the CA Access Control Command Line (see page 165)
seuidpgm Utility—Extract Trusted Programs (see page 216)

uxpreinstall Utility—Check for System Compliance

Valid on UNIX

The uxpreinstall utility verifies that a UNIX endpoint complies with UNIX Authentication Broker system requirements. uxpreinstall performs the following checks:

- Queries the operating system for the installed version, patches, libraries, and modules
- Resolves the domain name by querying the DNS server
- Searches for the LDAP and Kerberos services
- Uses the LDAP service to query Active Directory for information
- Scans for available ports
- Verifies the clock skew between the local host and the Active Directory domain
- Verifies that network applications, network servers, and ssh and sshd characteristics support Kerberized Single Sign On (SSO) login
- Discovers and displays all Global Catalog hosts in the domain

If the uxpreinstall utility finds a critical error that means it cannot perform subsequent checks, the utility stops immediately.

After uxpreinstall runs, it displays the result of the checks. Any errors or conflicts in the uxpreinstall output are issues that may cause UNIX Authentication Broker operational problems, for example, user authentication failure. We strongly recommend that you resolve any errors or conflicts that uxpreinstall identifies before you activate and use UNIX Authentication Broker.

Important! The uxpreinstall utility informs you of real or potential problems but does not correct them. You cannot use the utility to configure the operating system or UNIX Authentication Broker.

You can run uxpreinstall before or after you install UNIX Authentication Broker. If you run uxpreinstall before you install UNIX Authentication Broker, the utility creates a temporary Kerberos file and checks the configuration of the Kerberos file instead of the uxauth.ini configuration. If you run uxpreinstall after you install UNIX Authentication Broker, the utility does not create the temporary Kerberos file. Instead, it checks the value of the lookup_dc_list token in the [ad] section of the uxauth.ini file.

Note: To run uxpreinstall before you install UNIX Authentication Broker, copy the utility from another endpoint on which UNIX Authentication Broker is installed.

The following sections of the uxpreinstall output check if the endpoint configuration lets UNIX Authentication Broker users use Kerberized SSO login. If you do not want to enable SSO logins for UNIX Authentication Broker users, you can ignore any information in these sections:

- CHECKING KERBEROS RPMS
- CHECKING NATIVE KERBEROS
- == Reporting sshd characteristics affecting SSO operation ==
- == Reporting ssh characteristics affecting SSO operation ==
- Checks of network applications
- Checks of network servers

Note: For more information about using uxpreinstall to check system compliance, see the *Implementation Guide*.

This command has the following format:

```
uxpreinstall [-a user] [-w passwd] [-n ntp_server] [{-d domain | -s server}] [-p port]
[-f logfile] [-force] [-v level] [-l] [-h]
```

-a *user*

Defines the user account to use to log in to Active Directory.

Default: Administrator

-w *passwd*

Defines the password for the user account.

-n *ntp_server*

Defines the name of the Network Time Server (NTP).

-d *domain*

Defines the domain name where the Active Directory is installed.

-s *server*

Defines the name of the Active Directory server.

-p *port*

Defines the port number on which Active Directory listens.

-f *logfile*

Defines the name of the log file to use.

-force

Specifies to force continue the system compliance check regardless of errors

-v *level*

Defines the verbosity level of uxpreinstall output.

Options:

0—Displays a summary of the checks that uxpreinstall performs and any errors or conflicts that it identifies.

1—Displays the same information as 0 and additional information about each check.

2—Displays the same information as 1 and the commands that uxpreinstall uses for each check.

3—Displays the same information as 2 and the output of each command.

4—Displays the same information as 3 and extra information for some checks, for example, package details.

Default: 0

-l

Specifies to perform checks on the syslog file. Applicable for root users only.

-h

Specifies to display the utility help and exit.

Example: Run the uxpreinstall Utility

This example runs the uxpreinstall utility with the credentials of the administrator user against the Active Directory domain mydomain.com with a verbosity level of 1:

```
/opt/CA/uxauth/bin/uxpreinstall -a administrator -w admin -d mydomain.com -v 1
```

Example: The uxpreinstall Utility Report

The following is a snippet of the uxpreinstall utility report that shows how you determine whether your system complies with the system requirements:

```
OS detected: Linux 2.6.5-7.244-default
*********************************************
 CHECKING CLOCK SYNCHRONIZATION
*********************************************
Comparing the value of the currentTime attribute in DSE with the local time ...
Current clock skew is 34 sec.
The default value for the maximum clock skew is 300 seconds.
Warning! Significant clock skew can cause user authentication failure
---------------------------------------------
 W A R N I N G
---------------------------------------------

*********************************************
 CHECKING KERBEROS AUTHENTICATION VIA AD
*********************************************
principal_name = <Administrator@mydomain.com>

Kerberos authentication for <Administrator@mydomain.com> succeeded

---------------------------------------------
 S U C C E S S
---------------------------------------------

*********************************************
 CHECKING AD SCHEMA VERSION
*********************************************
Trying LDAP service at server.mydomain.com:389
Binding to Active Directory via 'server1.mydomaiin.com' ...

AD Schema version 31 (Windows Server 2003 R2 or Windows Server 7 (AD LDS))

supports full and partial UNAB integration modes.
---------------------------------------------
 S U C C E S S
---------------------------------------------
. . .
```

In this example, the output shows the following information:

- The operating system running on the local host—Linux 2.6.5-7.244-default
- The clock skew—34 seconds
- The Kerberos service—Kerberos authentication for <Administrator@mydomain.com> succeeded
- The Active Directory schema version—AD Schema version 31
- The operating system version where Active Directory is installed—Windows Server 2003 R2 or Windows Server 7
- The Active Directory schema supports both full and partial UNIX Authentication Broker integration modes

Services and Daemons in Detail

This section contains a complete alphabetic reference to all CA Access Control daemons and services.

CA Access Control Agent Manager

Valid on Windows

The CA Access Control Agent Manager service provides management services for the CA Access Control plugins. The CA Access Control Agent Manager service provides the plugins with the following services:

- Scheduling service—manages the plugins schedules.
- Watchdog service—verifies that the plugins are running and starts up plugins after failure.
- Messaging service—provides the plugins with message queue services and stores messages in case the Enterprise Management Server is unavailable.

The Agent Manager registry key contains registry entries to let you fine-tune the Agent Manager. You can find the key in the following location:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\common\AgentManager
```

CA Access Control Message Queue Service

Valid on Windows

The CA Access Control Message Queue service manages the Message Queue (TIBCO server) that handles all inbound and outbound messages between the Enterprise Management Server and other CA Access Control components. The Message Queue has a dedicated queue for each client component that communicates with the Enterprise Management Server, as follows:

- Report queue—Receives scheduled snapshots of the endpoint databases.

  The reporting service uses the snapshots to generate CA Access Control reports.

- Audit queue—Receives audit events that occur on the endpoints.

  You can configure CA Enterprise Log Manager to collect and report on the audit events.

- Server to endpoint queue—Receives data from the DMS that is collected by endpoints.

  For example, when you deploy a UNAB config policy the DMS sends the config policy to this queue. The UNAB agent then collects the policy from the queue and deploys the policy on the UNAB endpoint.

- Endpoint to server queue—Receives information from endpoints that is collected by the DMS.

  For example, a UNAB endpoint sends a heartbeat notification to this queue. The DMS then collects the heartbeat notification from the queue and updates the endpoint status in its database.

CA Access Control Web Service

Valid on Windows

The Web Service manages the web-based applications that you use to manage an enterprise installation of CA Access Control. The web-based applications are installed on the Application Server. The Application Server is installed by default on the Enterprise Management Server.

The Application Server contains the following web-based applications:

- CA Access Control Enterprise Management—Lets you manage policies across your enterprise and configure UNIX Authentication Broker endpoints. CA Access Control Enterprise Management also contains Privileged User Password Management (PUPM), which lets you manage privileged accounts across the enterprise and acts as a password vault for the privileged accounts.
- CA Access Control Endpoint Management—Lets you administer and configure individual CA Access Control endpoints through a central administration server.
- CA Access Control Password Manager—Lets you manage CA Access Control user passwords. You can modify the password of a CA Access Control user or force the user to change their own password when they next log in.

The WebService registry key contains registry entries to let you fine-tune the Web Service. You can find the key in the following location:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\WebService
```

Note: If you install the Enterprise Management Server on a UNIX computer, the eacws daemon manages the web-based applications.

CA Identity Manager - Connector Server (Java) Service

Valid on Windows

The CA Identity Manager - Connector Server (Java) Service manages communications with Java supported managed devices, such as Windows operating systems and SQL servers. This service also manages privileged accounts on Privileged User Password Management endpoints.

eacws Daemon

Valid on UNIX

The eacws daemon manages the web-based applications that you use to manage an enterprise installation of CA Access Control. The web-based applications are installed on the Application Server. The Application Server is installed by default on the Enterprise Management Server.

The Application Server contains the following web-based applications:

- CA Access Control Enterprise Management—Lets you manage policies across your enterprise and configure UNIX Authentication Broker endpoints. CA Access Control Enterprise Management also contains Privileged User Password Management (PUPM), which lets you manage privileged accounts across the enterprise and acts as a password vault for the privileged accounts.
- CA Access Control Endpoint Management—Lets you administer and configure individual CA Access Control endpoints through a central administration server.
- CA Access Control Password Manager—Lets you manage CA Access Control user passwords. You can modify the password of a CA Access Control user or force the user to change their own password when they next log in.

Note: If you install the Enterprise Management Server on a Windows computer, the CA Access Control Web Service manages the web-based applications.

KBLAudMgr Daemon—Session Logging

Valid on UNIX

The KBLAudMgr daemon manages the Keyboard Logger session recording agent. You use the Keyboard Logger to track privileged user sessions in UNIX and Linux endpoints. The Keyboard Logger records the interactive session, which you can replay when terminated and send to CA Enterprise Log Manager for analysis and reporting.

The [kblaudit] section of the seos.ini file contains tokens that let you fine-tune the Keyboard Logger agent.

PolicyFetcher Daemon

Valid on UNIX

The PolicyFetcher daemon regularly checks for deviations in the deployed policy, looks for deployment tasks on the DH, applies policy updates to the local CA Access Control database (seosdb), and sends a heartbeat to the DH at regular intervals.

Use the start DEVCALC selang command to start the deviation calculator. If you installed advanced policy management on the endpoint the PolicyFetcher runs the deviation calculator for you.

ReportAgent Daemon

Valid on UNIX

The ReportAgent daemon manages the ReportAgent that sends report snapshots and audit events to the Distribution Server for inclusion in CA Access Control, UNIX Authentication Broker, and CA Enterprise Log Manager reports. You run the ReportAgent utility from the *ACSharedDir*/bin directory on a UNIX computer, where *ACSharedDir* is the default directory /opt/CA/AccessControlShared. You can also use the report_agent.sh script to configure, start, and stop the ReportAgent.

The [ReportAgent] section of the accommon.ini file contains tokens that control the behavior of the Report Agent daemon.

ReportAgent Service (Windows)

Valid On Windows

The ReportAgent Service manages the ReportAgent that sends report snapshots and audit events to the Distribution Server for inclusion in CA Access Control, UNIX Authentication Broker, and CA Enterprise Log Manager reports. The ReportAgent Service automatically runs on start up if you installed CA Access Control on the endpoint and selected to install the ReportAgent.

The ReportAgent registry key contains registry entries to let you fine-tune the ReportAgent. You can find the key in the following location:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\ReportAgent
```

sepmdd Daemon (UNIX)

The Policy Model daemon.

The sepmdd daemon is the PMDB daemon. The sepmdd daemon performs the following functions:

- It administers the CA Access Control and UNIX databases of the Policy Model.
- It administers the subscribers' database.
- It propagates changes from the PMDB to the subscriber databases.

You can find the sepmdd daemon in the *ACInstallDir*/lbin directory. It starts the PMDB if it is already created.

Syntax

```
sepmdd policyModel
```

Parameters

policyModel

The name of the Policy Model.

Other Files

No other special files are used.

Notes:

When you use selang and choose a Policy Model as your target (using hosts pmd@hostname), queries to sepmdd apply to the PMDB but not to the various subscriber databases.

- Make sure that a PMDB does not become a subscriber of itself. If a PMDB is subscribed to itself, the Policy Model may block or the network may become overloaded, filling the disk in the process.
- When updating a Policy Model in the UNIX environment of selang, you can neither specify more than one user in the newusr command, nor specify more than one group in the newgrp command.
- When updating UNIX file attributes from selang, the Policy Model generates a message stating that the command was passed to its subscribers.
- When working on a Policy Model, you cannot query the status of UNIX file attributes.
- If you set the value of _shutoff_timeout_ to zero, the sepmdd daemon remains up and running indefinitely until you shut it off manually. Use the command sepmd -k to shut down the Policy Model daemon.

More information:

sepmd Utility (see page 184)
sepmdadm Utility—Create PMDB Definitions (see page 196)
seagent Daemon (see page 266)

How sepmdd Works

The CA Access Control agent, seagent, starts sepmdd; You do not need to run sepmdd explicitly. The sepmdd daemon runs under the logical user id "_seagent" for CA Access Control, and with the user id *root* in UNIX. You cannot designate another logical user under which sepmdd runs.

The PMDBs are stored in a common directory. You specify the name of the common directory with the _pmd_directory_ token in the [pmd] section of the seos.ini file, on the station where the Policy Models reside. Each Policy Model resides in a subdirectory of the common directory. The name of the Policy Model is the name of the subdirectory in which it resides.

When sepmdd starts, it checks whether any subscriber databases need updating, and updates them if necessary. After this startup process, sepmdd waits for user requests, which are sent by the Policy Model management program, sepmd, and by the selang utility, using seagent.

When sepmdd receives a request, it applies the request to the PMDB and sends the result back to the user. If the request should be propagated, sepmdd propagates the update to its subscriber databases.

The sepmdd daemon attempts to update a subscriber database for the period specified in the _QD_timeout_ token. If the maximum time elapses and the daemon does not succeed in updating a subscriber, it skips that particular subscriber and tries to update the remainder of the subscribers on its list. After it completes its first scan of the subscriber list, sepmdd then performs a second scan, in which it tries to update the subscribers that it did not succeed in updating during its first scan. During the second scan, it tries to update a subscriber until the connect system call times out (approximately 90 seconds).

Note: The _QD_timeout_ token may exist in both the seos.ini and pmd.ini files. If it does, sepmdd uses the value in the pmd.ini file.

If a subscriber is unavailable during the second scan, sepmdd attempts to send it updates every 30 minutes. To modify this interval, set the _retry_timeout_ token. Since the updates must be sent in the order in which they are received, sepmdd does not send subsequent updates to the subscriber database until it becomes available.

If you set the pull_option token in the [pmd] section of the subscriber database's seos.ini file to yes, the subscriber database is updated as soon as possible. seagent informs the parent Policy Models that the host is up for every Policy Model on the machine, and that its subscriber PMDBs are up, and sepmdd sends the update immediately.

Whenever sepmdd fails to update a subscriber database, it writes a warning message in the Policy Model error log. For more information about the Policy Model error log see the *Endpoint Administration Guide for UNIX*.

CA Access Control attempts to fully qualify subscribers as they are added or deleted from the Policy Model.

To remove a subscriber from the list of unavailable subscribers, enter the following command:

```
sepmd -r policyModel subscriber
```

If a subscriber database rejects an update, as can occur if the subscriber database differs from the PMDB, sepmdd writes an error message in the Policy Model error log and continues.

To view the error log, enter the following command on the host where the PMDB resides, enter:

```
sepmd -e policyModel
```

You can have sepmdd automatically shut itself down after a period of inactivity. By default, however, sepmdd does not shut itself down. If you want sepmdd to shut itself down, set the _shutoff_time_ token to a value greater than 0. This value indicates the minutes of inactivity allowed before sepmdd shuts itself down. To shut sepmdd down manually, enter:

```
sepmd -k policyModel
```

Important! Do *not* use the UNIX command *kill -9* to shut down sepmdd manually; this may destroy the PMDB.

UID/GID Synchronization

Because you may receive messages that refer to users by UID rather than by username, it is important to know each user's UID. But if you are using a PMDB and you pay no attention to how your new users' UIDs are assigned, the users may receive different UIDs on each subscriber machine. It is best to ensure, instead, that you can depend on each user to have the same UID everywhere; and the same is true of GIDs. See UID/GID Synchronization in the *Endpoint Administration Guide for UNIX*.

Filter Mechanism

You may want your PMDB to selectively update the subscriber stations below it. To define which records are sent to the subscriber stations, point the filter token in the pmd.ini file to a filter file. Updates to the subscriber stations are then limited to the records that pass the filter file.

A filter file consists of lines with six fields per line. The fields contain the following information:

- The form of access permitted or prohibited. The possible values are AUTHORIZE_DELETE, AUTHORIZE_MODIFY, CREATE, DELETE, DEPLOY, EDIT, FILESCAN, GET, SEOS_ACCS_READ, JOIN_DELETE, JOIN_MODIFY, MODIFY, READ, START, or UNDEPLOY.
- The environment affected. The possible values are AC, CONFIG, UNIX, NT, or NATIVE
- The class of the record. The possible values include all classes in CA Access Control, including user-defined classes.
- The objects within the class that the rule covers. For example, User1, AuditGroup, or TTY1
- The properties that the record grants or cancels. For example, OWNER and FULL_NAME in the filter line for user records means that any command having those user properties are filtered. You must enter each property exactly.
- Whether such records should be forwarded to the subscriber station or not. The possible values are PASS or NOPASS

You can use an asterisk in any field to mean “all possible values.” If more than one line covers the same records, the first applicable line is used.

In each line of the filter file, spaces separate the fields. In fields with more than one value, semicolons separate the values. Any line beginning with "#" is considered a comment line. Empty lines are not allowed. Here is an example of a line from a filter file:

CREATE	AC	USER	*	FULL-NAME;OBJ_TYPE	NOPASS
form of access	*environment*	*class*	*record name (* =all)*	*properties*	*treatment*

For example, suppose the file with this line is named TTY1_FILTER, and the pmd.ini file of the Policy Model TTY1 contains the line filter=/opt/CA/AccessControl//TTY1_FILTER. The Policy Model TTY1 does not send records that create new CA Access Control users with the FULL_NAME and OBJ_TYPE (Admin, auditor, and so on). The asterisk means "regardless of name."

The following are the selang commands that are relevant for each access value:

Access	selang Command
AUTHORIZE_DELETE	authorize-
AUTHORIZE_MODIFY	authorize
CREATE	newres, newusr, newgrp, newfile
DELETE	rmres, rmusr, rmgrp, rmfile, join- (UNIX)
DEPLOY	deploy
EDIT	editres, editusr, editgrp, editfile
FILESCAN	search
GET	get devcalc
JOIN_DELETE	join-
JOIN_MODIFY	join
MODIFY	chres, chusr, chgrp, chfile, join (UNIX)
READ	list
START	start devcalc
UNDEPLOY	deploy- (undeploy)

CA Access Control does not validate rules; therefore, if you enter an invalid value in a rule, the rule never matches an update transaction.

CA Access Control Policy Model Service (sepmdd)

Valid on Windows

CA Access Control Policy Model Service (sepmdd) is the PMDB service. It performs the following functions:

- Administers the CA Access Control and Windows databases of the Policy Model
- Administers the subscribers database
- Propagates changes from the PMDB to the subscriber databases

SeOSAgent starts the sepmdd service. There is no need to run sepmdd explicitly. The two possible states for each Policy Model are Started and Stopped.

The PMDBs are stored in a common directory. The registry value _pmd_directory_ in the subkey HKLM\Software\ComputerAssociates\AccessControl\Pmd specifies the name of the common directory. Each Policy Model resides in a subdirectory of the common directory. The name of the Policy Model is the name of the subdirectory in which it resides.

When sepmdd starts, it checks whether any subscriber databases need to be updated and, if necessary, updates them. After this startup process, the sepmdd service waits for user requests. User requests are sent by the Policy Model management utility sepmd and by selang using the CA Access Control Agent.

When a request is received, sepmdd applies it to the PMDB and sends the result back to the user. If the request should be propagated, sepmdd propagates the update to its subscriber databases.

The sepmdd service tries to update a subscriber database for 30 seconds. If this elapses and the service does not succeed in updating a subscriber, it skips that particular subscriber and tries to update the remainder of the subscribers on its list. After it completes its first scan of the subscriber list, sepmdd then performs a second scan, in which it tries to update the subscribers that it did not succeed in updating during its first scan. During the second scan, it tries to update a subscriber until the connect system call times out (approximately 90 seconds).

If a subscriber is unavailable during the second scan, sepmdd attempts to send it updates every 30 minutes.

Since the updates must be sent in the order in which they are received, sepmdd does not send subsequent updates to the subscriber database until it becomes available.

Each time sepmdd fails to update a subscriber database, a warning message is written in the Policy Model error log.

Filter Mechanism

You may want your PMDB to update the subscriber stations below it selectively. To define which records to be sent to the subscriber stations, set the registry key string value to a filter file. Updates to the subscriber stations are then limited to the records that pass the filter file.

Here is an example:

`HKEY_LOCAL_MACHINE\Software\ComputerAssociates\AccessControl\Pmd\`*`PolicyModelName`*`\Filter`

A filter file consists of lines with six fields per line. The fields contain this information:

The form of access permitted or prohibited

Valid values are: AUTHORIZE_DELETE, AUTHORIZE_MODIFY, CREATE, DELETE, DEPLOY, EDIT, FILESCAN, GET, SEOS_ACCS_READ, JOIN_DELETE, JOIN_MODIFY, MODIFY, READ, START, or UNDEPLOY.

The environment affected

Valid values are: AC, CONFIG, UNIX, NT, or NATIVE.

The class of the record

Valid values include all classes in CA Access Control, including user-defined classes.

The objects within the class that the rule covers

For example: User1, AuditGroup, or COM2.

The properties that the record grants or cancels

For example, including GROUPS and FULLNAME in the filter line for user records means that any command having those user properties is filtered. You must enter each property exactly as it appears.

Whether such records should be forwarded to the subscriber station

Valid values are: PASS, NOPASS

Note: You can use an asterisk to mean “all possible values” in any field. If more than one line covers the same records, the first applicable line is used.

In each line of the filter file, spaces separate the fields. In fields with more than one value, separate the values with semicolons. Any line beginning with “#” is considered a comment line. Empty lines are not allowed. Here is an example of a line from a filter file:

CREATE	AC	USER	*	FULLNAME;OBJ_TYPE	NOPASS
form of access	environment	class	record name (* =all)	properties	treatment

If, for example, the file with this line is named Printer1_Filter.flt and the registry key HKEY_LOCAL_MACHINE\Software\ComputerAssociates\AccessControl\Pmd\PM-\Filter contains the line "C:\Program Files\CA\AccessControl\\data\Printer1_Filter.flt," then Policy Model PM-1 will not send records that create new CA Access Control users with the FULLNAME and OBJ_TYPE (admin, auditor, and so on). The asterisk means "regardless of name."

The selang commands that are relevant for each access value are:

Access	selang Command
AUTHORIZE_DELETE	authorize-
AUTHORIZE_MODIFY	authorize
CREATE	newres, newusr, newgrp, newfile
DELETE	rmres, rmusr, rmgrp, rmfile, join- (UNIX)
DEPLOY	deploy
EDIT	editres, editusr, editgrp, editfile
FILESCAN	search
GET	get devcalc
JOIN_DELETE	join-
JOIN_MODIFY	join
MODIFY	chres, chusr, chgrp, chfile, join (UNIX)
READ	list
START	start devcalc
UNDEPLOY	deploy- (undeploy)

Note: CA Access Control does not validate rules; therefore, if you enter an invalid value in a rule, the rule will never match an update transaction.

Registry Subkeys

Each PMDB has its own registry subkey under:

```
HKEY_LOCAL_MACHINE\Software\ComputerAssociates\AccessControl\Pmd
```

This subkey contains the values that define and determine the activity of the PMDB. The sepmdd utility creates a subkey, if it does not already exist, with the minimum number of entries needed.

Notes

- When you use selang and choose a Policy Model as your target (using hosts pmd@hostname), queries to sepmdd apply to the PMDB but not to the various subscribers' databases.
- Ensure that a PMDB does not become a subscriber of itself. If a PMDB is subscribed to itself, the Policy Model may block or the network may become overloaded, filling the disk in the process.
- You cannot specify more than one user with the newusr command when you are working in the UNIX environment using selang to update a Policy Model.
- You cannot specify more than one group in the newgrp command when you are working in the UNIX environment using selang to update a Policy Model.
- When updating UNIX file attributes from selang, the Policy Model generates a message stating that the command has been passed to its subscribers.
- When working on a Policy Model, you cannot query the status of Windows file attributes.
- The sepmdd service remains active indefinitely until deactivated with the -k options.

More information:

seagent Daemon (see page 266)
sepmd Utility (see page 184)
sepmdadm Utility—Create PMDB Definitions (see page 196)

seagent Daemon

Valid on UNIX

The seagent daemon accepts requests from remote stations, and applies them to the local CA Access Control and UNIX databases, or to the PMDBs. It also checks that the Watchdog daemon seoswd is running, and if it is not, restarts it.

Note: When you load CA Access Control (seload) it also starts seagent; this daemon does not work independently and cannot be started using the seagent command.

Seagent waits for connections on the seoslang and seoslang2 TCP services (whose default values are 8890 and 8891 respectively). When a connection request arrives, seagent forks a child process to handle the communication on the connection, and continues waiting for new connections.

The child processes of seagent get the requests from the client, and apply them to the local database.

The Agent is also responsible for the following:

- Updating the UNIX user file /etc/passwd, the system's shadow password file, and the UNIX group file /etc/group
- Alerting Policy Model daemons when an update is sent
- Alerting the parent Policy Models of both the local host and any Policy Model on the machine when a subscriber station (that has been down) is available for updating

CA Access Control uses only ports 8890 and 8891. We recommended that you do not change these ports.

The seagent agent uses the RPC mechanism and therefore the portmapper must be running on the local machine. For additional information on the portmapper, check your system documentation.

This command has the following format:

```
seagent
```

More information:

seoswd Daemon (see page 273)
sepmdd Daemon (UNIX) (see page 258)

seauxd Daemon

Valid on UNIX

The seauxd daemon is the CA Access Control auxiliary daemon, which manages Unicenter calendar updates.

To activate seauxd, set the TNG_calendars configuration setting to yes.

The seauxd daemon is started by the seosd daemon according to initialization settings. The seauxd daemon performs the following functions:

- Analysis requests from seosd
- Unicenter TNG calendar retrieval-To activate this function set TNG_calendars token in the [seauxd] section of the seos.ini file to yes. When this function is activated, seosd sends the list of Unicenter TNG calendars to seauxd. The seauxd daemon calls Unicenter TNG, updates the status of each calendar, and returns an updated list of calendars to seosd.

The [seauxd] section of seos.ini contains of number tokens to allow you to fine tune the seauxd daemon.

seosd Daemon

Valid on UNIX

The CA Access Control authorization daemon. The executable file seosd is the main CA Access Control daemon. A daemon is a process that has disconnected from both its controlling TTY and its parent process. The CA Access Control daemon makes the runtime decisions required to grant or deny access to a resource.

Only root can invoke seosd, and only a user with the ADMIN or OPERATOR attribute can shut it down.

The CA Access Control daemon opens, reads, and updates the database. No other process can access this database while the CA Access Control daemon is running. The CA Access Control daemon also blocks any write, delete, or rename access to critical files, such as the CA Access Control audit and trace files and, optionally, the CA Access Control binary files.

The seosd executable becomes a daemon only if one or both of the following conditions is true:

- The trace messages are not sent to the screen; that is, you set the trace_to token in the seos.ini file to *file*, *file,stop*, or *none*.
- You specify no argument (except-d) on the command line when invoking the utility.

If none of these conditions are true, seosd remains a regular process, connected to the terminal from which you invoked it.

During startup, seosd also invokes the following processes:

- seagent, the CA Access Control agent daemon.
- seoswd, the CA Access Control watchdog daemon.

The CA Access Control daemon is completely initialized only after these daemons are also running. After initialization, these three daemons maintain a type of handshaking protocol to ensure they are all alive and responding. If one of these daemons is found to be absent, one of the other two daemons automatically restarts it.

This command has the following format:

```
seosd [-d|argument]
```

Note: If you enter seosd with no arguments, it runs seosd as a daemon.

argument

Ignored. However, if you specify an argument, seosd remains a regular process.

-d

Runs seosd as a daemon and forces tracing to the trace_file.

selogrcd Daemon—Collect Audit Records

Valid on UNIX

Collector daemon for the CA Access Control log routing system.

Note: selogrcd does not work in IPv6-only environments.

The CA Access Control log routing daemons, selogrd and selogrcd, provide system administrators with convenient, selective access to the audit log records.

The selogrcd utility is the collection daemon. This daemon collects the selected audit log records sent by various satellite systems and stores them in the audit collection file. The default file is *ACInstallDir*/log/seos.collect.audit.

Two tokens enhance audit collection file management. Both tokens are in the [selogrd] section of the seos.ini file

- Use the Caudit_size token to specify the maximum size of the audit collection file. When the file reaches this size, CA Access Control creates a backup file and opens a new file.
- Use the CbackUp_Date token to specify a automatic backup interval and timestamp for the audit collection file.

You can force selogrcd to start a new audit file by sending it a USR1 signal. Once you have the selogrcd process ID, send it a USR1 signal using a kill command such as:

```
kill -USR1 processID
```

When it receives a USR1 signal, selogrcd renames the existing audit file to *ACInstallDir*/log/seos.collect.bak and creates a new audit file. You can also use a cron job to perform this task periodically. A sample script that performs this task is provided in the directory *ACInstallDir*/samples/selogrcd.

Note: You can expand the functionality of the selogrcd daemon by writing programs at your site that use the APIs provided with CA Access Control. For more information, see the *SDK Guide*.

This command has the following format:

```
selogrcd [-d] [-l lock-file-name]
```

-d

Specifies the debug mode. In this mode, selogrcd does not become a daemon. It sends debug information to the terminal.

-h

Displays the help for this utility.

-l *lock-file-name*

Specifies the name of the lock file to be used (*lock-file-name*). By default, selogrcd uses the file *ACInstallDir*/lock/selogrcd.

Note: If you set selogrd to work on a different log file (such as a PMDB log file), the lock file has an extension based on the PMDB name or the data file name that was used as the parameter for the selogrd command (see page 270).

selogrd Daemon—Emit Audit Records

Valid on UNIX

Emitter daemon for the CA Access Control log routing system.

Note: selogrd does not work in IPv6-only environments.

The CA Access Control log routing, daemons selogrd and selogrcd, provide system administrators with convenient, selective access to the audit log records.

The selogrd utility is the emitter daemon. This daemon distributes selected local audit log records to the various destination hosts; reformats audit log records into email messages, ASCII files, or user windows; and sends out notification messages based on audited events.

Note: The CA Access Control daemon must be up and running before the log routing daemons can collect any meaningful information on CA Access Control events. If the CA Access Control daemon is not running, selogrd routes only old audit records.

The log routing daemons use a configuration file to determine where each audit log record is sent, the format in which the log record is written, and which records are routed. By default, selogrd uses the *ACInstallDir*/log/selogrd.cfg audit log route configuration file. The names of the configuration file and other global environment variables that selogrd and selogrcd use are specified in the CA Access Control initialization file, seos.ini.

The selogrd daemon periodically restarts and reads the configuration file. In addition, you can force the selogrd daemon to restart at a specified time. To do so, you must send the following HUP signal:

```
kill -HUP processID
```

processID

Defines the selogrd process ID. (Use the UNIX ps command to find it; see your UNIX documentation for more information.)

The selogrd utility provides API access for programmers working under CA Access Control. The Logroute API allows programmers to incorporate their own options into the CA Access Control audit log system to support in-house alerts not provided by the current log-routing facility. The Logroute API also allows programmers to use the log routing daemons to provide functions to their own programs. For more information on all the CA Access Control APIs, see the *SDK Developer Guide*.

This command has the following format:

```
selogrd [-audit fileName] [-config fileName] [-d] \
    [-data fileName] [-pmdb policy-model-name]
```

-audit *fileName*

Defines the audit file to use instead of the file listed in seos.ini for the input audit file.

-config *fileName*

Defines the configuration file to use instead of the file listed in seos.ini for the configuration file.

-d

Specifies to print debug messages.

-data *fileName*

Defines the data file to use instead of the file listed in seos.ini to store routing progress information.

-h

Displays the help for this utility.

-pmdb *policy-model-name*

Instructs selogrd where to route audit data from a PMDB. The command tells selogrd to send audit data from the PMDB that you specified in the command, to the audit file that you specified in the audit_log token in the pmd.ini file of the PMDB.

By default, selogrd uses the data file and lock file that consist of the Policy Model name. If you specify the data file or lock file or both on the command line, those files override the default values. The lock file and data file names should be different from those of the selogrd that route the audit data of the station. selogrd can only support Policy Model names of 12 characters.

The audit data that is sent from a PMDB appears in the collected audit file as if it comes from a station with the name policy-model-name@station-name

More information:

The Audit Log Route Configuration File selogrd.cfg (see page 403)

seostngd Daemon

Valid on UNIX

The CA Access Control synchronization daemon for Unicenter TNG.

Unicenter Security and CA Access Control together manage the administration of your enterprise IT environment before total migration occurs. To reduce the complexity of using different product tools to perform administrative tasks, we are providing a synchronization daemon.

This daemon is called seostngd. CA Access Control sends Policy Model database (PMDB) updates through CA Common Communication Interface (CAICCI) to seostngd. The daemon listens for updates on CAICCI and then translates the messages into equivalent cautil commands to update the Unicenter Security database with this global data.

Current Unicenter TNG processing can still update other Unicenter TNG client installations. You must run seostngd on the same machine as the Unicenter Security database (normally referred to as the Unicenter master machine.) CA Access Control should also be running on the same machine.

This command has the following format:

```
seostngd

seostngd {-stop|-shut}
```

seoswd Daemon

Valid on UNIX

The CA Access Control watchdog daemon.

The watchdog (seoswd) monitors the file information and digital signatures of programs that are defined in the database as trusted programs. Monitoring is performed in the background with a minimal load on the system. The CA Access Control agent daemon seagent automatically starts seoswd.

The seoswd daemon performs the following functions:

- It monitors the programs that you defined in the PROGRAM class of the database. If the watchdog detects that a program was modified, it notifies the CA Access Control daemon, seosd, which marks the program as untrusted. The seosd daemon does not allow an untrusted program to run. The seosd daemon also marks the program's status change to untrusted in the database and creates an audit record.
- It monitors files that are defined as secured files. These files are defined in the SECFILE class in the database.
- It monitors seosd to ensure it is running. If the watchdog detects a problem with seosd, it automatically restarts it.
- The seoswd daemon uses the system log syslogd to notify the security administrators when it detects that seosd has stopped responding. All system log messages are submitted as AUTH facility. For more information on the system log facility, see your system man pages under the syslogd and syslog.conf sections.
- It reports several events to CA Access Control, and creates audit records for programs and secured files that were found to be altered.
- It allows you to specify interval and fixed scanning schedules for trusted programs and secure files.
- The watchdog ignores any signal except SIGHUP; you cannot kill the seoswd daemon unless you first shut down seosd. However, if you execute the command kill -SIGHUP *pid*, the watchdog scans all trusted programs and secure files in the database.

There are two ways in which you can set up the Watchdog scanning mechanism:

1. Determine a start time and then repeat scans at a given interval.

   For example, when checking trusted programs, the Watchdog will start the first scan at *PgmTestStartTime* and will check all the trusted programs. Rescanning will take place *PgmTestInterval* seconds after the beginning of the previous scan.

2. Scan at given times.

Note: In both cases, the Watchdog will sleep periodically for a predetermined rest period *(PgmRest* seconds) during each scan. The Watchdog rests in order to prevent system overload.

You can choose to use one mechanism or both simultaneously. For example, starting at 12:00, scan every 4 hours as well as at 13:00 and 17:30.

In addition to the above mentioned mechanisms for routine scanning of the trusted programs and secured files, there is a way to perform a one-time scan on demand by sending a HUP signal (see token SignalMinInterval).

If you invoke seoswd without an argument, it runs as a daemon. If you invoke seoswd with the -d argument, it runs as a daemon, but displays all debug information on the terminal from which you invoked it.

More information:

seuidpgm Utility—Extract Trusted Programs (see page 216)

Chapter 3: Configuration Files

This section contains the following topics:

The accommon.ini File

The accommon.ini configuration file contains tokens that control the initialization process of the Report Agent and tokens that control general communication settings, for example, UNIX Authentication Broker registration settings with CA Access Control Enterprise Management. The accommon.ini file is divided into the following sections:

Section	Description
communication	Contains tokens that control general communication settings
global	Contains CA Access Control global settings
ReportAgent	Contains tokens that control the Report Agent settings
AccountManager	Contains token that control the Account Manager settings

communication

In the [communication] section, the tokens control the communication and encryption options.

Distribution_Server

Defines the Distribution Server URL. You can define more than one Distribution Server in a comma-separated list.

Example: tcp://ds.comp.com:7222, tcp://ds_dr.comp.com:7222

Default: none

endpoint_to_server_queue

Defines the name of the message queue that the endpoint uses to send information to CA Access Control Enterprise Management.

Default: ac_endpoint_to_server

jvm_gc

Defines the optional parameter for tuning the Garbage Collector. The JVM uses this parameter while communicating with the Distribution Server.

Values: A string representing complete GC tunables.

Example: jvm_gc = -XX:+UseConcMarkSweepGC -XX:+UseParNewGC

Default: empty value

jvm_ms

Defines the optional parameter for the JVM initial heap size -Xms. The JVM uses this parameter while communicating with the Distribution Server.

Values: The size is measured in MB.

Example: jvm_ms = 512

Default: 0

jvm_mx

Defines the optional parameter for a Java virtual machine (JVM) maximum heap size -Xmx. The JVM uses this parameter while communicating with the Distribution Server.

Value: The size is measured in MB.

Example: jvm_mx = 1024

Default: 0

jvm_ps

Defines the optional parameter for the JVM permgen size -XX:MaxPermSize. The JVM uses this parameter while communicating with the Distribution Server.

Values: The size is measured in MB.

Example: jvm_ps = 128

Default: 0

server_to_endpoint_broadcast_queue

Defines the name of the message queue that CA Access Control Enterprise Management uses to broadcast messages to all endpoints.

Default: ac_server_to_endpoint_broadcast

server_to_endpoint_queue

Defines the name of the message queue that the Enterprise Management Server uses to send messages to the endpoint and authenticate using the reportserver user.

Default: server_to_endpoint_queue

server_to_server_broadcast_queue

Defines the name of the message queue that the Enterprise Management Server uses to broadcast topics and authenticates using the reportserver user.

Default: ac_server_to_server_broadcast

server_to_server_queue

Defines the name of the message queue that the Enterprise Management Server uses to send messages and authenticate using the reportserver user.

Default: ac_server_to_server

server_to_endpoint_queue

Defines the name of the message queue that CA Access Control Enterprise Management uses to send messages to the endpoint.

Default: ac_server_to_endpoint

ServerVersion

Defines the Distribution Server version for forward compatibility.

Example: 12.01.0648

Default: none

ssl_custom

Specifies whether to use the host name verifier function.

Limits: 0, do not use the host name verifier function; 1, use the host name verifier function

Default: 0

ssl_hostname

Defines the SSL host name.

Default: none

ssl_identity

Defines the identity of the Report Agent.

Limits: The full pathname to a file containing the certificate data.

Default: none

ssl_issuer

Defines issuer certificates to the SSL connection.

Limits: The full pathname to a file containing the certificate data.

Default: none

ssl_key

Defines the Report Agent private key.

Limits: The full pathname to a file containing the private key.

Default: none

ssl_noverifyhost

Specifies whether to enable verification of the host certificate.

Limits: 0, disable host certificate verification; 1, enable host certificate verification

Default: 0

ssl_noverifyhostname

Specifies whether to enable verification of the host name.

Limits: 0, disable host name verification; 1, enable host name verification

Default: 0

ssl_trace

Specifies whether to enable SSL tracing.

Limits: 0, disable SSL tracing; 1, enable SSL tracing

Default: 0

ssl_trusted

Defines trusted certificates to the SSL connection.

Limits: The full pathname to a file containing the certificate data.

Default: none

global

In the [global] section, the tokens control the behavior of the CA Access Control endpoint.

accommon_path

Specifies the full path name of the accommon directory.

Default: /opt/CA/AccessControlShared/

AC_Version

Defines the version of CA Access Control installed on the endpoint.

Default: none

java_home

(Linux s390) Defines the path to the Java libraries.

Example: For an IBM J2SE version 5.0 JRE installed on a Linux390 computer: /opt/ibm/java2-s390-50/jre

Default: none

ReportAgent

In the [ReportAgent] section, the tokens control the behavior of the Report Agent daemon (ReportAgent).

audit_enabled

Specifies whether you want to send endpoint audit data to the Distribution Server.

Values: 0—no; **1**—yes

Default: 0

audit_filter

Defines the full pathname to the file that contains filtering rules for audit records that the Report Agent routes to an external source (such as CA Enterprise Log Manager). This file determines which records the Report Agent routes.

Default: *ACSharedDir*/etc/auditrouteflt.cfg

audit_queue

Defines the name of the queue to which the Report Agent sends endpoint audit data.

Default: queue/audit

audit_read_chunk

Defines the maximal audit records the Report Agent tries to collect in a single read of the audit files.

Limits: A positive integer.

Default: 300

audit_send_chunk

Defines the maximal audit records that the Report Agent sends to the Distribution Server in each connection. When the number of audit records the Report Agent collects reaches this number it sends these records to the Distribution Server.

Limits: A positive integer

Default: 1800

audit_sleep

Define the length of time the Report Agent sleeps between generating audit reports.

Limits: A positive integer representing a number of seconds.

Default: 10

audit_timeout

Defines the cycle at which the Report Agent must send endpoint audit data to the Distribution Server. If this amount of time passes from the last send, the Report Agent sends audit data to the Distribution Server even if the number of records it collected is less than the audit_send_chunk value.

Limits: A positive integer representing a number of seconds.

Default: 300

Debug

Specifies whether the Report Agent logs debug information.

If you specify yes (1), the Report Agent logs the following:

- CA Access Control reports to *ACSharedDir*/log/ac2xml.log
- UNIX Authentication Broker reports (uxauthd) to *ACSharedDir*/log/unab2xml.log
- CA Access Control audit reports that are sent to CA Enterprise Log Manager to *ACSharedDir*/log/ac2elm.log

- UNIX Authentication Broker audit reports that are sent to CA Enterprise Log Manager to *ACSharedDir*/log/unab2elm.log
- Keyboard Logger reports that are sent to CA Enterprise Log Manager to *ACSharedDir*/log/kbl2elm.log

Limits: 0, Report Agent does not log debug information; 1, Report Agent logs debug information

Default: 0

elm_event_interval

Defines the interval in seconds, at which the Report Agent sends user sessions audit events to CA Enterprise Log Manager.

Limits: 0; no interval, send audit events when messages size exceeds the value specified in the elm_max_msg_size token; any positive integer.

Default: 60

elm_max_msg_size

Defines the maximum size of Keyboard Logger messages, in bytes, that the Report Agent sends to CA Enterprise Log Manager.

Value: Any positive integer

Default: 300000

interval

Defines the interval, in minutes, at which CA Access Control generates and sends reports to the Distribution Server.

The *schedule* setting defines the interval start time and the days it operates on. If the Report Agent starts later than a scheduled occurrence, it sends a report at the next calculated interval (from the schedule) and then at the defined intervals after that on scheduled days.

Example: If you have schedule=8:30@Mon,Tue,Wed and interval=5 and the Report Agent loads on Tuesday at 8:47 am, the Report Agent generates and sends a report at 8:50 am. This is the earliest cycle calculated from the scheduled start using the 5 minute interval.

Values: 0—No interval (use scheduled occurrences only); *positive integer*—number of minutes to use as interval

Default: 0

reportagent_enabled

Specifies whether reporting is enabled (1) on the local computer.

Default: 0

schedule

Defines when reports are generated and sent to the Distribution Server.

You specify this setting in the following format: time@day[,day2][...]

For example, "19:22@Sun,Mon" generates reports every Sunday and Monday at 7:22 pm.

Default: 00:00@Sun,Mon,Tue,Wed,Thu,Fri,Sat

send_queue

Defines the name of the reporting queue on the Distribution Server to which the Report Agent sends snapshots of the local database and any PMDBs.

Default: queue/snapshots

restart_enabled

Specifies restart of the ReportAgent daemon. Specify 1 to enable the restart.

Default: 0

More information:

auditrouteflt.cfg File—Filter Audit Records Routing (see page 396)

AccountManager

In the [AccountManager] section, the tokens control the behavior of the Account Manager plug-in.

INSTALL_ACCOUNT_MNG

Specifies whether to configure Account Management on the endpoint.

Values: Yes, No

Default: No

Example: INSTALL_ACCOUNT_MNG="no"

Note: Configure the Distribution Server parameters to activate the AccountManager.

Interval

Specifies the AccountManager plug-in interval in seconds

Default: 300

Note: Applicable if you set the ScheduleType control value to 2.

JCS_USER_DN

Specifies the Java Connector Server (JCS) administrator user DN.

Values: DN format string.

Default: cn=root,dc=etasa

Example: JCS_USER_DN="cn=root,dc=etasa"

JCS_USER_PSSWD

Specifies the JCS administrator password.

Values: any string.

Default: no default value.

Note: The JCS_USER_PSSWD value is replaced by asterisks (*) after installation.

JCS_SERVER_DN

Specifies the JCS server DN.

Values: DN format string.

Default: dc=im,dc=etas

Example: JCS_SERVER_DN="dc=im,dc=etasa"

JCS_SERVER_PORT

Specifies the JCS port.

Values: Port number

Default: 20411

Example: JCS_SERVER_PORT=20411

JCS_SSL

Defines the JCS communication protocol.

Values: 'yes' for SSL connection, otherwise 'no'.

Default: yes

Example: JCS_SSL="yes"

OperationMode

Defines whether the AccountManager plug-in is enabled or disabled.

Options: 1, plug in enabled, 0, plug in disabled

Default: 1

PluginPath

Defines the full pathname of the AccuntManager plug-in.

Default: /opt/CA/AccessControlShared/lib/AccountManager.so

QueryFilter

Specifies a custom value to add to the Message Queue receive queue filter.

Options:

- "ENDPOINT_CUSTOM1="
- "ENDPOINT_CUSTOM2="
- "ENDPOINT_CUSTOM3="
- "ENDPOINT_CUSTOM4="
- "ENDPOINT_CUSTOM5="
- "ENDPOINT_OWNER="
- ENDPOINT_DEPARTMENT="

Default: no value

Note: You can use more than one custom property, using the AND operand.

Example: "ENDPOINT_DEPARTMENT='Finance' AND 'ENDPOINT_CUSTOM1=Accounting'"

Important! When specifying the custom property verify that:

- You use apostrophes to specify the property value.
- You use the AND, OR operands when specifying more than one property.
- You use the parenthesis when using the OR operand.

Schedule

Specifies the AccountManager plug-in a schedule string.

Default: 00:00@Sun,Mon,Tue,Wed,Thu,Fri,Sat

Note: Applicable if you set the ScheduleType control value to 3.

ScheduleType

Defines the AccountManager plug-in scheduling type.

Options:

- 0—Run once
- 1—Run on demand
- 2—Run every N seconds
- 3—Run according to the scheduling string: 00:00@Sun.Mon,tue,Wed,Thu,Fri,Sat

Default: 2

The kblaudit.cfg—Filter Keyboard Logger Audit Records

Valid on UNIX

The kblaudit.cfg file filters audit records on a host by defining records that sent to the audit file. Each line represents a rule for filtering out audit information. The filter rules you configure apply to the kbl.audit file.

By default, the kblaudit.cfg file is located in the following directory:

```
/opt/CA/AccessControl/etc
```

The kblaudit.cfg file contains two sections, [EXCLUDE] and [INCLUDE] to help you filter keyboard logger audit records. Each section contains entries that represent a filter rule.

Example: The kblaudit.cfg filter sections

The following snippet of the kblaudit.cfg file is an examples of how you edit the kblaudit.cfg [EXCLUDE] and [INCLUDE] sections:

```
[EXCLUDE]
TRACE;*;*;test_user; test_user; test_user;*;*seos.ini*
[INCLUDE]
TRACE;*;*; test_user; test_user; test_user;*;*AccessControl*
```

In this example, you excluded from the kbl.audit file audit records from seos.ini that the user test_user performed and to include records that the user test_user performed in Access Control.

Use the kblaudit.cfg file to filter out records in the following audit event types, each type by a different syntax:

- login events (see page 285)
- trace message on a user (see page 286)

Note: A * in any column in each type of syntax stands for "any value".

Kblaudit.cfg—Login Events Filter Syntax

Valid on UNIX

Audit records that belong to a login event have the following filter format:

```
LOGIN;UserName;UserId;TerminalName;LoginProgram
```

Login

Specifies that the rule filters user trace records.

UserName

Defines the name of the accessor.

UserId

Defines the native user ID of the accessor.

TerminalName

Defines the remote host name at which the event occurred.

LoginProgram

Defines the name of the program that attempted to log in or out.

Limits: cmdlog

kblaudit.cfg —Trace Messages On User Events Filter Syntax

Valid on UNIX

Audit records that belong to a trace message on a user event have the following filter format:

TRACE;*TracedClassName;TracedObjectName;RealUserName;ACUserName;AuthorizationResult;TraceMessageMask;KBLSessionID*

TRACE

Specifies that the rule filters user trace records.

TracedClassName

Defines the name of the object class the user tried to access.

Options: KBL raw, KBL output, KBL input, KBL execargs

TracedObjectName

Defines the name of the object that the user tried to access.

RealUserName

Defines the name of the logged in user that generated the trace records.

ACUserName

Defines the name of the effective user that generated the trace record.

AuthorizationResult

Defines the authorization result.

Values: P (permitted), D (denied), *

TraceMessageMask

Defines the trace message that was generated.

KBLSessionID

Displays the keyboard logger sessions ID

The seos.ini Initialization File

Valid on UNIX

The seos.ini file contains various setup and initialization tokens used by CA Access Control. Each token occupies a line in the file, in the following format:

token = *value*

The lines containing the tokens for a particular utility, daemon, or other facility of CA Access Control are grouped together in sections. Each section starts with a header line that gives the section name inside square brackets. Every token belongs to a section. For example, the following line starts the section that governs the serevu utility:

```
[serevu]
```

The seos.ini file, as installed, is protected by CA Access Control and cannot be updated while CA Access Control is running. The file, as defined by default in CA Access Control, has READ access because many utilities access this file during their processing. If they cannot read the seos.ini file, they will fail.

Enter the following selang command to let an authorized user update the file while CA Access Control is running:

```
newres FILE /opt/CA/AccessControl//seos.ini owner(authUser)
```

where *authUser* is the name of an authorized user. This command establishes that *authUser* is the owner of the file, and as the owner of the file, *authUser* can always update it.

You can use CA Access Control Endpoint Management or the seini utility to read, add, modify, and delete tokens in initialization files.

Note: The seini utility can only update the seos.ini file when seosd is *not* running, or when a rule in the database specifically permits it.

Using the *secons -rl* command, you can reload an seos.ini file with updated tokens without having to restart the seosd daemon.

The following table lists all the sections in the seos.ini file.

Section	Description
AccountManager	Multiple JCS endpoint module
AgentManager	CA Access Control plugins management
crypto	Cryptographic module library settings.
daemons	A list of CA Access Control daemons the seload utility runs automatically.
Dependency	A list of products that use CA Access Control as an embedded component, as defined by users.
devcalc	Policy deviation calculator (devcalc) settings.
kblaudit	Keyboard logging session tracking settings.
lang	CA Access Control management interface (selang) settings.
ldap	LDAP server settings for the LDAP sample exit.
logmgr	Logging facility settings.
message	Message file settings.
mfsd	Mainframe synchronization daemon (mfsd) settings.
OS_user	Enterprise user store usage settings.

Section	Description
package	A list of installed CA Access Control packages.
pam_seos	Pluggable Authentication Module (PAM) programming interface settings.
passwd	Password replacement and user-related services settings.
pmd	Common Policy Model database settings.
policyfetcher	Policy fetcher daemon (policyfetcher) settings.
PUPMAgent	Privileged User Password Management daemon (pupmagent) settings.
seagent	seagent daemon settings.
seauxd	Auxiliary daemon (seauxd) settings for Unicenter calendar updates.
segrace	User login information utility (segrace) settings.
seini	Configuration file management utility (seini) attributes.
selock	Desktop inactivity protection utility (selock) settings.
selogrd	Log routing daemons (selogrd and selogrcd) settings.
seos	Global configuration settings.
SEOS_syscall	SEOS_syscall kernel module settings.
seosd	Authorization daemon (seosd) settings.
seosdb	Database checking and rebuilding settings.
seoswd	Watchdog daemon (seoswd) settings.
serevu	Unsuccessful login attempts resolution utility (serevu) utility settings.
sesu	CA Access Control switch user utility (sesu) settings.
sesudo	CA Access Control substitute user do utility (sesudo) utility settings.
standalone	Standalone computer administration settings.
tcp_communication	Common TCP connection settings.
tng	CA Access Control integration with Unicenter settings.

crypto

In the [crypto] section, the tokens control aspects associated with the cryptography module.

ca_certificate

Defines the full pathname to the Certificate Authority (CA) certificate database.

Default: *ACInstallDir*/data/crypto/def_root.pem

communication_mode

Specifies whether secure socket layer (SSL) protocols are enabled.

If you set this to ssl_only, only SSL V2, SSL V3, and TLS connections are enabled. This means that this computer cannot communicate with computers that do not support SSL, and so cannot communicate with computers that are running versions of CA Access Control earlier than r12.0, which do not support SSL.

Note: Computers that are running CA Access Control r12.0 and later do support SSL.

If the fips_only token is set to 1, the actual communication mode is set to ssl_only in FIPS mode (that is, TLS), and the communication_mode token is ignored.

Valid values are:

- all_modes
- ssl_only
- non_ssl

Default: non_ssl

CAPKIHOME

Defines the installation directory of CAPKI.

Default: /opt/CA/SharedComponents/CAPKI

encryption_methods

Specifies the encryption libraries that the CA Access Control Agent uses to decrypt messages. The Agent attempts to use each library in the list, in turn, until the decryption is successful.

Limits: libaes256, libaes192, libaes128, libdes, libtripledes, libscramble

Default: libaes256, libaes192, libaes128, libdes, libtripledes

fips_only

This token controls whether CA Access Control works in FIPS only mode. In this mode all non-FIPS functions are disabled.

Valid values:

1 CA Access Control works in FIPS only mode

0 CA Access Control works in non-FIPS mode

Default: 0

LIBRARY_PATH

Defines the directory for the ETPKI cryptographic library.

private_key

Defines the full pathname to the subject private key.

Default: *ACInstallDir*/data/crypto/sub.key

ssl_port

Defines the port for SSL communications between CA Access Control clients and services.

Default: 5249

subject_certificate

Defines the full pathname to the subject certificate.

Default: *ACInstallDir*/data/crypto/sub.pem

daemons

In the [daemons] section, each token specifies whether (and if so, how) the seload utility executes a particular program from the CA Access Control installation directory. Each token name corresponds to either a CA Access Control daemon name or is a program nickname and can be assigned several values.

program-name

Specifies one of two possibilities:

- The name of a daemon or other program to be matched with:
  - a *yes* value, so that seload runs the program with default parameters
  - a *no* value, so that seload does not run the program
  - a set of parameters, so that seload runs the program with those parameters

  For example, enter the following to run serevu from the CA Access Control installation directory with default parameters:

  ```
  serevu=yes
  ```

  Enter the following to refrain from running serevu; this is the same as using no serevu token at all.

  ```
  serevu=no
  ```

  Enter the following to run serevu from the CA Access Control installation directory with the specified parameters:

  ```
  serevu=-f 3 -d 6m -t 1m -s 5m
  ```

- A dummy string, to be matched with the absolute path name of a daemon or other program, followed by optional parameters, so seload runs the program accordingly.

  For example, enter the following to run the serevu utility that resides in the /opt/CA/AccessControl//bin directory, with the specified parameters:

  ```
  run_it=/opt/CA/AccessControl//bin/serevu -f 3 -d 6m -t 1m
  ```

  To include specifications for several programs, use the token once for each program.

Default: no

Note: You do not need to specify the seosd daemon. seload always ensures that the seosd daemon is running.

seload_wait_timeout

Specifies the time (in seconds) that seload waits until the main process is running.

Default: 45

Dependency

In the [Dependency] section, each user-defied token specifies a product that uses CA Access Control as an embedded component.

product-name

Specifies a product that uses CA Access Control as an embedded component. Valid values are:

0 - Not an embedded product

1 - An embedded CA Access Control product.

Default: No default products specified.

devcalc

In the [devcalc] section, the tokens control aspects associated with the policy deviation calculator.

dms_command_retry_interval

Defines the number of seconds between each DMS notification command retry.

Default: 60

init_ac_db

Obsolete.

max_dms_command_retry

Defines the maximum number of times the policy deviation calculator retries to send update notifications to the DMS before giving up.

Default: 3

max_lines_request

Defines the maximum number of lines (from the policy deviation data file) that the *get devcalc* selang command returns at any one time. You then need to retrieve additional lines using the following command:

```
get devcalc params("offset=X")
```

X

Defines the line offset returned by the previous *get devcalc* output.

Default: 50

kblaudit

The tokens in the [kblaudit] section control the behaviour of the Keyboard Logger session tracking program.

audit_back

Specifies the name of the Keyboard Logger backup audit log file.

Default: *ACInstallDir*/log/kbl.audit.bak

audit_group

Specifies the group that can read the audit logs. If you set this token to **none**, only root can read the audit logs. CA Access Control does not verify the value of this token, so if you enter an invalid group name, CA Access Control does not assign any group permissions to the audit log files.

To change the group ownership of an existing audit log file, complete the following steps:

Use the selang command chgrp to set the group ownership of the files.

Change the UNIX permissions by entering the following command:

```
chmod 640 ACInstallDir/log/seos.audit
```

Default: none

audit_log

Specifies the name of the Keyboard Logger audit log file.

Default: *ACInstallDir*/log/kbl.audit

audit_max_files

Specifies the maximum number of audit log files to keep in backup mode. When reached, CA Access Control deletes the earliest backup file when the latest file is created.

Limits: a positive integer.

Default: 0

Note: When set to 0, CA Access Control accumulates backup files and does not delete earlier files.

audit_size

Specifies the maximum size, in KB, of the audit log file.

Minimum value: 50 KB.

Default: 24000

Note: CA Access Control stops writing audit records to the audit file when the audit file size exceeds 2 GB.

BackUp_Date

Specifies the criterion by which CA Access Control backs up the audit log file, and if CA Access Control adds a timestamp to the backup file name.

CA Access Control *always* backs up the audit log file when it reaches the size specified in the audit_size configuration setting.

Values: none, yes, daily, weekly, monthly

- yes—CA Access Control backs up the audit log file when it reaches the size specified in audit_size and adds a timestamp to the backup file name.
- none—CA Access Control backs up the audit log file when it reaches the size specified in audit_size and does not add a timestamp to the backup file name.
- daily, weekly, monthly——CA Access Control backs up the audit log file whenever the specified interval has elapsed *and* when it reaches the size specified in audit_size, and adds a timestamp to the backup file name. However, if no audit events are written to the audit log file in the specified interval, CA Access Control does not back up the file after the interval elapses.

  Note: CA Access Control counts the specified interval from the time that it creates the first audit log file, and backs up the file at midnight on the appropriate day.

Example: The configuration setting has a value of weekly and CA Access Control creates the audit log file at 9:00 a.m. Friday 1 April. Many audit events occur this week and the audit log file exceeds the audit_size configuration setting on Monday 4 April. CA Access Control backs up the audit log file on 4 April and adds a timestamp to the backup file name. A week after the audit log file was first created, at midnight Friday 8 April, CA Access Control again backs up the audit log file and adds a timestamp to the backup file name.

Default: NONE

cmd_log

Specifies the link to the Keyboard Logger cmdlog binary file.

Default: /etc/AC

error_back

Specifies the name of the Keyboard Logger error log backup file.

Default: *ACInstallDir*/log/kbl.error.bak

error_group

Specifies the group that can read the error log files. If you set this token to **none**, only root can read the error log files. CA Access Control does not verify the value of this token, so if you enter an invalid group name, CA Access Control does not assign any group permissions to the error log files.

To change the group ownership of an existing error log file, complete the following steps:

Use the selang command chgrp to set the group ownership of the files.

Change the UNIX permissions by entering the following command:

```
chmod 640 ACInstallDir/log/seos.audit
```

Default: none

error_log

Specifies the name of the Keyboard Logger error log file.

Default: *ACInstallDir*/log/kbl.error

error_size

Defines the maximum size, in KB, of the error log file.

Limits: A minimum value of 50 KB.

Default: 500

kbl_enabled

Specifies whether the Keyboard Logger is enabled.

Values: yes, no

Default: no

kbl_flush_timeout

Specifies the user session inactivity interval, in seconds, after which the printable logged data is stored in the kbl audit file. Set the token to 0 to disable.

Default: 30

Kbl_seos_trace

Specifies whether seosd activates trace on session and sends user activity data to the Keyboard Logger.

Values: yes, no

Default: yes

OS_etc_shells

Specifies the name of the operating system shells file.

Default: /etc/shells

socket_name

Specifies the socket name for the Keyboard Logger audit manager.

Default: *ACInstallDir*/kblserver

lang

In the [lang] section, the tokens specify the attributes used by the selang command language programs: selang, Security Administrator, and seadm.

check_password

Determines whether selang will request users to specify their own passwords. Valid values include:

no-selang does not require any passwords

yes-Users are prompted to enter their passwords.

Default: no

exit_timeout

Specifies the maximum time, in seconds, that CA Access Control allows the exit program to execute. After this time has passed, CA Access Control kills the exit program.

Default: 30

exits_dir

Specifies the target directory where exits are installed by the *ACInstallDir*/lbin/install_exits.sh shell script.

Default: *ACInstallDir*/exits

exits_source_dir

Specifies the source directory of the exits to be installed by the *ACInstallDir*/install_exits.sh shell script.

Default: *ACInstallDir*/samples/exits-src

help_path

Specifies the directory in which lang help files are located.

Default: *ACInstallDir*/data/langhelp

language

Defines the language CA Access Control installs in (for internal use).

Default: english

max_groups_buffsize

Specifies the buffer size, in KB, that the security administrator uses when communicating with the database. This token is used when a UNIX update needs to be applied.

Default: 128

no_check_password_users

Specifies users who are not asked to enter their passwords.

This token is relevant only if the token check_password is set to **yes**.

Valid values include a list of users separated by commas.

Default: none

passwd_copy

Specifies how the machine password file (/etc/passwd) or PMDB password file *(/PMDB_Directory*/policies/pmdb/passwd) is updated when you copy the temporary file back to the original after changing user information.Valid values include:

fast_copy - Copies information over the file.

rename - Changes the directory to point to the new file.

Default: fast_copy

post_group_exit

Specifies the path of the exit program to be called after a group command is executed in the UNIX environment.

Default: *ACInstallDir*/exits/lang_exit.sh

post_user_exit

Specifies the path of the exit program to be called after a user command is executed in the UNIX environment.

Default: *ACInstallDir*/exits/lang_exit.sh

pre_group_exit

Specifies the path of the exit program to be called before a group command is executed in the UNIX environment.

Default: *ACInstallDir*/exits/lang_exit.sh

pre_user_exit

Specifies the path of the exit program to be called before a user command is executed in the UNIX environment.

Default: *ACInstallDir*/exits/lang_exit.sh

query_size

Specifies the maximum number of records to be listed in a database query.

Default: 100

RecvTimeOut

Specifies the maximum time, in seconds, that selang will wait to receive information before timing out.

If you set the value to 0, there will be no time-out.

Default: 60

SendTimeOut

Specifies the maximum time, in seconds, that selang will wait to send information before timing out.

If you set the value to 0, there will be no time-out.

Default: 60

SetBlockRun

Specifies whether to check if a program is trusted and block the execution of untrusted programs. The execution blocking is performed regardless whether the program is a setuid or a regular program.

Valid values include the following:

yes-All programs defined with viapgm authorization rules have the blockrun property set to yes.

no-All programs defined with viapgm authorization rules have the blockrun property set to no.

suid-All setuid programs have the blockrun property set to yes, and all other programs have the blockrun property set to no.

Default: yes

swap_deletion_order

Defines the order in which the "ru *userName* unix" command (user deletion) is executed in selang. Normally, this command is first executed in the AC environment, and then in the UNIX environment. In some cases (for example, a group administrator deleting a user) where you would want to reverse this order.

Valid values are:

no - remove the user from the AC environment before the UNIX environment.

yes - remove the user from the UNIX environment before the AC environment.

Default: no

timeout

Specifies the maximum time, in seconds, the client waits for seosd daemon to respond. If seosd does not respond within this period, an error message is sent noting that seosd is not responding. The client then stops trying to connect to seosd.

Default: 90

use_old_commands

Specifies whether to disable old ACF2™ compatibility commands (ag, lg, rg, lu, au, and so on).

Limits: 0—do not support old commands, 1—support old commands

Default: 1 (support old commands)

use_unix_file_owner

Specifies whether a UNIX owner of a file can define the file to CA Access Control. If the value is yes, an owner of a file in UNIX can define it to CA Access Control, using the newres or newfile command.

If the file is already defined to CA Access Control, the user cannot change its parameters in the database unless the user is allowed to do so according to the normal CA Access Control authorization rules.

Valid values are yes and no.

Default: no

ldap

In the [ldap] section, the tokens specify the attributes used to locate the LDAP server and input data. These parameters are used only by the ldap sample exit located in *ACInstallDir*/samples/ldap/exits/S50CREATE_Ldap_u.sh.

base_entry

Specifies the point in the LDAP directory tree to be used as the base entry point.

For example, you may use o=*organization_name*, c=*country_name*.

Default: Token not set

host

Specifies the host name of the LDAP server.

Default: Token not set (localhost)

path

Specifies the LDAP client base directory.

Default: Token not set (/usr/local/ldap)

port

Specifies the LDAP server port (optional)

Default: Token not set (389)

logmgr

In the [logmgr] section, the tokens control the behavior of the logging facility.

audit_back

Specifies the name of the audit log backup file. Only CA Access Control can write to this file. Users can have READ access only to this file.

Default: *ACInstallDir*/log/seos.audit.bak

audit_group

Specifies the group that can read the audit logs. If you set this token to **none**, only root can read the audit logs. CA Access Control does not verify the value of this token, so if you enter an invalid group name, CA Access Control does not assign any group permissions to the audit log files.

To change the group ownership of an existing audit log file, complete the following steps:

Use the selang command chgrp to set the group ownership of the files.

Change the UNIX permissions by entering the following command:

```
chmod 640 ACInstallDir/log/seos.audit
```

Default: none

audit_log

Specifies the name of the audit log file. When this file reaches the size specified in *audit_size*, CA Access Control closes the file, renames it with the name in *audit_back*, and creates a new audit log. Only CA Access Control can write to this file. Users can have READ access only to this file.

Default: *ACInstallDir*/log/seos.audit

audit_max_files

Defines the maximal number of audit log backup files CA Access Control accumulates when it performs date-triggered backups. When the BackUp_Date configuration setting is set to anything other than *none*, CA Access Control continuously accumulates date-triggered backup files. This configuration setting lets you reduce disk space CA Access Control uses for audit log backups. When the number of audit log backup files reaches the limit you set, CA Access Control deletes the oldest backup file when it creates the newest.

Values:

- **0**—keep all audit log backup files.
- *n*—a positive integer greater than zero.

Note: You cannot remove redundant audit log backup files manually because CA Access Control protects these automatically. Also, if the audit reporting is enabled, CA Access Control does not delete a backup file until the Report Agent finishes processing it.

Default: 0

audit_size

Specifies the maximum size, in KB, of the audit log file.

Minimum value: 50 KB.

Default: 10240

Note: CA Access Control stops writing audit records to the audit file when the audit file size exceeds 2 GB.

BackUp_Date

Specifies the criterion by which CA Access Control backs up the audit log file, and if CA Access Control adds a timestamp to the backup file name.

CA Access Control *always* backs up the audit log file when it reaches the size specified in the audit_size configuration setting.

Values: none, yes, daily, weekly, monthly

- yes—CA Access Control backs up the audit log file when it reaches the size specified in audit_size and adds a timestamp to the backup file name.
- none—CA Access Control backs up the audit log file when it reaches the size specified in audit_size and does not add a timestamp to the backup file name.

- daily, weekly, monthly——CA Access Control backs up the audit log file whenever the specified interval has elapsed *and* when it reaches the size specified in audit_size, and adds a timestamp to the backup file name. However, if no audit events are written to the audit log file in the specified interval, CA Access Control does not back up the file after the interval elapses.

  Note: CA Access Control counts the specified interval from the time that it creates the first audit log file, and backs up the file at midnight on the appropriate day.

Example: The configuration setting has a value of weekly and CA Access Control creates the audit log file at 9:00 a.m. Friday 1 April. Many audit events occur this week and the audit log file exceeds the audit_size configuration setting on Monday 4 April. CA Access Control backs up the audit log file on 4 April and adds a timestamp to the backup file name. A week after the audit log file was first created, at midnight Friday 8 April, CA Access Control again backs up the audit log file and adds a timestamp to the backup file name.

Default: NONE

error_back

Specifies the name of the error log backup file.

Default: *ACInstallDir*/log/seos.error.bak

error_group

Specifies the group that can read the error log files. If you set this token to **none**, only root can read the error log files. CA Access Control does not verify the value of this token, so if you enter an invalid group name, CA Access Control does not assign any group permissions to the error log files.

To change the group ownership of an existing error log file, complete the following steps:

Use the selang command chgrp to set the group ownership of the files.

Change the UNIX permissions by entering the following command:

```
chmod 640 ACInstallDir/log/seos.audit
```

Default: none

error_log

Specifies the name of the error log file. When this file reaches the size specified in *error_size*, CA Access Control closes the file, renames it with the name in *error_back*, and creates a new error log. Only CA Access Control can write to this file.

Default: *ACInstallDir*/log/seos.error

error_size

Defines the maximum size, in KB, of the error log file.

Limits: A minimum value of 50 KB.

Default: 50

irecorder_audit

Specifies whether the IR API library routes audit events of existing PMDs in addition to the local security daemon audit events.

"all" - routes audit events of Policy Models in addition to the local security daemon audit events.

"localhost" - routes audit events of the local security daemon only.

Default: all

logconnected

Prevents TCP-CONNECTED records from being written to the audit log.

Set logconnected to No to use this feature.

Default: no

More information:

seerrlog Utility—Display Error Log Records (see page 157)

message

In the [message] section, the tokens control the behavior of the message utility semsgtool.

filename

Specifies the location and name of the file that supplies most of the messages that appear in response to typed selang commands.

Default: *ACInstallDir*/data/seos.msg

MessagesDirectory

Specifies the location of the CA Access Control messages file.

Default: *ACInstallDir*/data/msg

mfsd

In the [mfsd] section, the tokens define the mainframe synchronization daemon options.

mfsd_trace_file

Specifies the location of the file to which CA Access Control mainframe synchronization daemon mfsd trace messages are written.

If this token is set to **no**, the trace file is not created.

Default: *ACInstallDir*/log/mfsd.trace

OS_User

The tokens in the [OS_User] section define the settings used by CA Access Control for enterprise users and enterprise groups.

create_user_in_db

Specifies whether CA Access Control creates an XUSER record for a user who is not defined to CA Access Control, when that user logs in.

Note: This setting applies only if you use enterprise users (osuser_enabled is set to 1).

Limits: yes, no

Default: yes

nonunix_unabgroup_enabled

Specifies whether CA Access Control supports non UNIX groups of users in the UNIX Authentication Broker database.

Limits: yes, no

Default: no

nonunix_ldapgroup_enabled

Specifies whether CA Access Control supports non UNIX groups of users, located on LDAP servers.

Limits: yes, no

Default: no

osuser_enabled

Specifies whether enterprise users and groups are enabled.

Limits: yes, no

Default: yes

UserCache_groups_max

Defines the maximum number of groups in the runtime user cache table.

Default: 1000

UserCache_max

Defines the maximum number of entries in the runtime user cache table.

Default: 20000

UserCache_timeout

Defines the interval (in minutes) before a record is removed from the runtime user cache table.

Default: 60

verify_osuser

Specifies whether CA Access Control verifies that a user exists in an enterprise store before it creates an enterprise user record (XUSER) in CA Access Control.

Limits: no, CA Access Control lets you create an enterprise user record only if that user is defined in the enterprise user store; yes, CA Access Control always lets you create an enterprise user record.

Default: no

package

In the [package] section, the tokens specify the packages you selected to install.

Client, Server, Admin, Mfsd, Tng, Stop, Api

Indicates whether you selected to install the specified package.

Default: no

pam_seos

In the [pam_seos] section, the tokens help you to more fully exploit the programming interface PAM (Pluggable Authentication Module).

api_update_lastaccterm

Specifies whether the API libraries update the last access time and date of a user (via SEOS_VerifyCreate).

Valid values are:

0 - the last access time and date is not updated.

1 - the last access time and date is updated.

Default: Token not set (0)

bypass_services

Defines which services PAM bypasses.

Default: ftp,vsftpd

call_segrace

Specifies whether to automatically call the segrace utility with any login.

Valid values are yes and no.

Default: no

call_sepass

Specifies whether to use the sepass utility in the pam_seos password management service.

Values: No, Yes

Default: Token not set (No)

debug_mode_for_user

Specifies whether to inform the user of the reason for login denial.

Valid values are yes and no.

Default: no

failed_login_file

Specifies the location of the failed login audit file pam_seos.

Default: *ACInstallDir*/pam_seos_failed_logins.log

pam_login_events_enabled

Specifies whether pam_seos sends login events to seosd.

Values: **0** - do not send login events; **1** - send login events

Default: 1

pam_get_groups

Specifies whether pam_seos attempts to retrieve user groups from operating system.

Values: 0 - do not attempt to retrieve groups; 1 - attempt to retrieve groups

Default: 1

pam_groups_timeout

Defines the timeout interval, in seconds, that CA Access Control PAM uses for API to retrieve user groups.

Default: 10

PamPassUserInfo

Specifies whether pam_seos sends user information to seosd. This is required when you use enterprise users, which CA Access Control has no information for. Set this setting to 0 if you are not using enterprise users (osuser_enabled = no).

Values: 0 - do not send user information; **1** - send user information.

Default: 0

pam_surrogate_events_enabled

Specifies whether pam_seos sends surrogate events to seosd.

Values: **0** - do not send surrogate events; **1** - send surrogate events.

Default: 1

process_failed_logins

Specifies whether pam_seos calls pam_authenticate to authenticate user passwords and process failed logins.

Set this setting to 0 if you do *not* want pam_authenticate to be called twice.

Values: 0 - do not call pam_authenticate from CA Access Control PAM module; **1** - call pam_authenticate from CA Access Control PAM module.

Default: 1

serevu_use_pam_seos

Specifies whether serevu should use the pam_seos login failure log file instead of the system file.

This feature increases the accuracy of serevu.

Default: *yes* on HP-UX Itanium (IA64) and Linux, *no* on all other operating systems

passwd

In the [passwd] section, the tokens define password replacement and other user-related services.

AllowedGidRange

Specifies the range of GIDs that the user can add, update, and delete. Values outside this range represent reserved GIDs that CA Access Control cannot update.

Note: If only one integer is specified, all integers between one and the specified integer are reserved GIDs. If you specify a number that is higher than the upper limit, the default upper limit is applied (30000). If you specify a negative number, the default lower limit is applied (1). The applied lower limit for any number is +1 of the specified lower limit. For example, if *AllowedGidRange = 100, 3000*, then 101 is treated as the lower limit.

Limits: -1 to 2147483647

Default: 100,30000

AllowedUidRange

Specifies the range of UIDs that the user can add, update, and delete. Values outside this range represent reserved UIDs that CA Access Control cannot update.

Note: If only one integer is specified, all integers between one and the specified integer are reserved UIDs. If you specify a number that is higher than the upper limit, the default upper limit is applied (30000). If you specify a negative number, the default lower limit is applied (1). The applied lower limit for any number is +1 of the specified lower limit. For example, if *AllowedUidRange = 100, 3000*, then 101 is treated as the lower limit.

Limits: -1 to 2147483647

Default: 100,30000

AllowRootProp

Specifies whether root password changes made using sepass -p or sepass -s are sent to the Policy Model. The PMD then propagates the password to its subscribers.

Valid values are yes and no.

Default: no

change_pam

Specifies whether the local host uses PAM for password authentication and changes in the LDAP database.

Default: no

Check_Adm_Rules

Specifies whether to enforce password rules for ADMIN and PWMANAGER users.

Default: no

Check_All_User_Rules

Specifies whether selang should check the Password Rules for all the users.

Valid values are yes and no.

If this token is set to yes, selang checks the Password Rules for all the users.

If this token is set to no, selang checks the Password Rules only for the user who changes the password.

Default: no

Note: This token is supported when using the API only.

CreateHashedPasswdDatabase

(DEC UNIX only). Specifies whether an exit script runs after each CA Access Control command that creates, updates or removes a user record, or after each user password changed with the sepass utility.

Note: For more usage instructions, see the README file in *ACInstallDir*/samples/exits-src/USER_POST directory.

Default: no

DefaultHome

Specifies the default home directory of the system. The user's home directory is a subdirectory of the specified system home directory. For example, if the system home directory is /home, the new user's home directory is /home/*username*. If specified, the value for this token overrides the value in the client's lang.ini file. If you specify *nohomedir* then a home directory is not automatically set.

Default: /home

DefaultPasswdCmd

Specifies the default password program. If specified, this password program is used when sepass is started and seosd is not running.

Default: /bin/passwd

DefaultPgroup

Specifies the primary group that CA Access Control assigns to a new UNIX user if no value is entered.

Default: other

DefaultShell

Specifies the default shell that CA Access Control assigns to a new UNIX user if no value is entered. If specified, the value for this token overrides the value in the client's lang.ini file.

Default: /bin/sh (or /sbin/sh on HP-UX)

Dictionary

Defines the full pathname of the file containing the words that *cannot* be used as passwords.

Note: To use this file, you must set the dictionary format password rule (use_dbdict) to *file* and set UseDict setting to *yes*. If the dictionary format is set to *db*, passwords that cannot be used are taken from the CA Access Control database and this setting is ignored. This is the default on UNIX.

Important! This token is obsolete. Use dictionary in the database instead.

Default: /usr/dict/words

GeneratePasswd

Specifies whether sepass generates a new password by itself.

Valid values are yes and no.

If you set this token to **no**, the user is asked to enter a new password.

Default: no

HomeDirUpd

Specifies whether CA Access Control updates the group ownership of the user's home home directory when the user's primary group changes.

Valid values are **yes** and **no**

Default: yes

nis_env

Specifies whether the local host is an NIS or NIS+ client.

Valid values are no, nis, or nisplus.

Default: no

NisPlus_server

Specifies whether this station is an NIS+ server.

Valid values are yes and no.

If token value is yes, CA Access Control treats password replacements as NIS+ password replacements.

Default: no

only_local

Determines whether the default setting for sepass includes the -l flag.

Valid values are yes and no.

If this token is set to yes, sepass will replace the password only in the local; that is, in the local password file (usually /etc/passwd), security files, and the local database.

Default: no

only_pmdb

Specifies whether the default setting for sepass includes the -p flag. If token value is yes, it instructs sepass to change the password only on the PMDB at the host specified.

If no such database is defined, sepass does nothing.

Default: no

passwd_distribution_encryption_mode

Specifies which method is used to encrypt user passwords when passwords are distributed as part of the Policy Model service.

Valid values are:

1 - Compatibility mode, to distribute passwords between CA Access Control systems that do not use long passwords (This includes all machines running pre-r12.0 versions of CA Access Control.)

2 - MD5 mode, to distribute passwords between CA Access Control systems that use long passwords and are also running Linux.

3 - Bidirectional mode, to distribute passwords securely, as clear text within encrypted messages, between any CA Access Control systems that use long passwords.

Default: 1

passwd_format

Indicates whether the password changes are propagated to an NT host.

Setting this token to **NT** means that one of the hosts you are administering is an NT host.

Default: none

passwd_local_encryption_method

Specifies which method is used to encrypt user passwords when storing these passwords locally.

Valid values are:

crypt - The standard one-way UNIX encryption that uses only the first eight characters of the password (as a DES key). Specifying crypt disables the use of long passwords.

md5 - MD5 hash function that can encrypt passwords of indefinite length. Specifying md5 enables the use of long passwords.

Default: crypt

PromptOldPassword

Specifies whether to prompt local users for their old password when sepass is invoked through /opt/CA/AccessControl//bin/segrace. (You must use the full path).

Setting this token to **yes** indicates that the users are prompted for their old passwords.

Default: yes

quiet_mode

Specifies whether sepass displays a copyright notice and a message about propagating passwords to Policy Models.

Default: no

RootPwAsOwn

Specifies whether sepass lets a privileged user change the root password as if changed by root (using the *-x* option).

Valid Values are:

yes-Privileged users can use sepass to change the root password as if changed by root. They cannot change the root password as themselves (administrative change).

no-Privileged users can use sepass to change the root password only as themselves (administrative change).

For example, a privileged user can use the following command to change the root password if this token is set to *yes*:

```
sepass -x root
```

The same user cannot use the following command to change the root password:

```
sepass root
```

If this token is set to *no*, the opposite is true.

Default: no

SaveGroupAttrs

Specifies whether the previous group file owner, group, and mode are preserved after an update of a group in the UNIX environment.

Valid values are yes and no.

If you set this token to **no**, new values are set to 0, 0, 644 respectively.

Default: no

SavePasswdAttrs

Specifies whether the previous password file owner, group, and mode are preserved after an update of a user in the UNIX environment.

Valid values are yes and no.

If you set this token to **no**, new values are set to 0, 0, 644 respectively.

Default: no

Shadow_Admin_Change

(AIX platforms only). Specifies whether the ADMCHG flag gets added to the user entry in the /etc/security/passwd file when an administrator changes the password from selang or using sepass.

Default: no

UIDAlgorithm

Specifies which free UID algorithm to employ when adding new users. Setting it to any other value would select the older process. The *new* algorithm provides for UID numbers over 4 KB and is faster.

Default: new

UseDict

Specifies whether to use the dictionary file (set with the Dictionary setting) when verifying a password.

Note: To use the dictionary file, you must also set the dictionary format password rule (use_dbdict) to *file*. If the dictionary format is set to *db*, passwords that cannot be used are taken from the CA Access Control database and this setting is ignored.

Default: no

YpGrpCmd

Specifies the command to use for generating the NIS group map.

Default: make group

YpMakeDir

Specifies the name of the makefile directory to be used when creating NIS maps.

Default: /var/yp

YpPassCmd

Specifies the command to use for generating the NIS password map.

Default: make passwd

YpServerGroup

Specifies the group file from which the NIS group map is made.

Default: /etc/group

YpServerPasswd

Specifies the password file from which the NIS password map is made.

Default: /etc/passwd

YpServerSecure

Specifies the name of the security file containing passwords that is used for building the NIS password map.

Default: Varies by platform:

- IBM AIX: /etc/security/passwd
- HP-UX: /.secure/etc/passwd
- Sun Solaris: /etc/shadow

YpTimeOut

Specifies the time, in seconds, that a new client (selang, Security Administrator, and so forth) can run the ypbind test, which determines whether the local host is connected to a NIS server. At expiration, the client exits and an error message appears.

The default value of zero (**0**) means that no ypbind test is conducted.

Default: 0

More information:

sepass Utility—Set or Replace a Password (see page 182)

pmd

In the [pmd] section, the tokens determine the PMDB attributes.

Note: In addition to seos.ini, each policy model has a configuration file named pmd.ini.

_min_retries_

Specifies the minimum number of attempts that sepmdd should make to resend the next queued update to an unavailable subscriber. The sepmdd loops through the list of subscribers for outstanding updates and increments the counter each time it cannot resend the update to an unavailable subscriber. The subscriber is marked unavailable after the minimum number of attempts specified in this token.

Default: 4

_pmd_backup_directory_

Defines the directory that CA Access Control uses to store Policy Model backups. CA Access Control stores each PMD backup in a subdirectory named *pmd_name*.

Default: *ACInstallDir*/data/policies_backup

_pmd_directory_

Specifies the directory in which the PMDBs reside. The name can contain up to 70 alphanumeric characters. Specify the full path of the directory. Each Policy Model resides in the directory *pmdDirectory*/*pmdName*.

Default: *ACInstallDir*/policies

_PMD_DIRECTORY_

Same as _pmd_directory_

_PMD_EXEC

Defines the name of the Policy Model daemon.

_QD_timeout_

Specifies the maximum time, in seconds, that the sepmdd daemon waits while attempting to update a subscriber database during the first scan of its subscriber list. If the time elapses and the daemon does not succeed in updating a subscriber, it skips that particular subscriber and tries to update the remainder of the subscribers on its list.

After completing the first scan of the subscriber list, sepmdd then performs a second scan in which it attempts to update the subscribers it did not succeed in updating during the first scan. During the second scan, it tries to update a subscriber until the connect system call times out (approximately 90 seconds).

Default: 3

_retry_timeout_

Specifies the time, in minutes, to wait before trying to resend an update to an unavailable subscriber, after the minimum number of attempts specified in _min_retries_ has been made. It marks the subscriber available after the number of minutes defined by this token elapses.

A subscriber is marked unavailable until:

- It is manually released.
- sepmdd is manually shutdown and restarted. The sepmdd is restarted if:
  - if a language facility attempts to connect to it.
  - if a parent PMDB wants to send an update.
  - the pull option is triggered by a subscriber. This optionally occurs when CA Access Control starts on the subscriber.
- The pull option is triggered by the unavailable subscriber.

Note: Shutting down sepmdd too often is not desirable because it takes time to restart the daemon, which results in slowing the whole propagation process. Allowing it to be on all the time is also undesirable because there maybe some stability issues, but it is only a conjecture.

Default: 30

_shutoff_time_

Specifies the time, in minutes of activities before sepmdd quits. If the token value is zero, sepmdd never quits.

Default: 0

ClientOperationTimeout

Defines the timeout period, in seconds, a client waits for a response from the Policy Model.

Default: 60

is_maker_checker

Specifies whether to use Dual Control.

Valid values are yes and no.

If the token value is **yes**, you cannot update the database directly, but only through a PMDB, and two administrators-a Maker and a Checker-must collaborate on the update.

Default: Token not set (no)

pass_auth

Specifies whether sepass verifies the invoker's password during a remote password change. The sepass utility always compares the old password the user enters with the password stored in the local prodname database. If you set this token to yes, sepass also compares the old password the user running sepass enters with their own password as it is stored in the remote prodname database (usually pmdb). This means that the sepass user must enter their own password even when changing the password for another user.

Values: yes, no

Default: yes

pull_option

Specifies whether subscriber databases are updated as soon as they become available.

Valid values are yes and no.

If the token value is **yes**, seagent sends a message to the parent Policy Models of both the local host and any Policy Model on the machine as soon as the subscriber station becomes available. sepmdd then updates the subscriber immediately, instead of waiting for the next half-hourly retry.

Default: yes

send_unix_env

Specifies whether the sepmd -n option sends the contents of the policy model password files and group files.

Valid values are yes and no.

yes-The *sepmd -n* option sends the contents of the policy model password files and group files.

no-The *sepmd -n* option does **not** send the contents of the policy model password files and group files.

Default: yes

ShutdownWaitingTimeout

Defines the timeout period, in seconds, the Policy Model waits for its components to gracefully shut down. If the Policy Model components did not shut down gracefully, the Policy Model shuts down forcefully.

Default: 60

synch_uid

Specifies whether CA Access Control forces subscribers to use the same uid as the parent Policy Model host when creating a new UNIX user.

updates_in_chunk

Define the maximum number of commands that the Policy Model sends to each of its subscribers in each cycle of a loop.

Default: 10

More information:

sepmd Utility (see page 184)

policyfetcher

In the [policyfetcher] section, the tokens control the behavior of the policy fetcher daemon (policyfetcher).

check_deployment_tasks

Defines how often, in seconds, policyfetcher checks for new deployment tasks (DEPLOYMENT resources) on the Distribution Host.

Default: 3600 (every 10 minutes)

Limits: A minimum value of 60

deploy_timeout

Defines the number of seconds policyfetcher waits for a deployment or undeployment task to complete on the endpoint.

Default: 900

devcalc_command

Defines the selang command that policyfetcher uses to run the deviation calculation.

Default: start DEVCALC params(-nonotify)

Example: start DEVCALC params(-nonotify -precise)

dh_command_retry_interval

Defines the number of seconds between each DH notification command retry.

Default: 60

endpoint_heartbeat

Defines the frequency at which policyfetcher sends a heartbeat to the Distribution Host (DH). The frequency is a factor of the check_deployment_task setting, and determines how many times policyfetcher checks deployment tasks before it sends a heartbeat. For example, if check_deployment_task is set to the default 600 seconds (10 minutes) and you set this to 6, policyfetcher sends a heartbeat every 3600 seconds (1 hour).

After sending the heartbeat, the policyfetcher also runs the deviation calculator (start devcalc command) and then waits 60 seconds for the deviation calculation to complete. After 60 seconds, policyfetcher continues to check that local endpoint information is identical to DH information.

Default: 6

max_deployment_errors

Defines the maximum number of deployment errors that the endpoint sends to the DMS.

Default: 10

max_dh_command_retry

Defines the maximum number of times policyfetcher retries to get update notifications from DH before giving up.

Default: 10

max_dh_retry_cycles

Defines the maximum number of cycles policyfetcher retries to get update notifications from production DHs before moving to disaster recovery DHs.

Default: 5

policy_verification

Specifies whether the policyfetcher daemon verifies new deployment tasks on a backup CA Access Control database before executing the tasks.

Valid values:

1 - Run policy verification

0 - Disable policy verification

Default: 0

policyfetcher_enabled

Specifies whether to run the policyfetcher daemon.

Valid values:

1 - Run policyfetcher

0 - Disable policyfetcher

Default: 0

PUPMAgent

In the [PUPMAgent] section, the tokens determine the functionality of the Privileged User Password Management Agent.

EnableLogonIntegration

Specifies that terminal integration is enabled.

Limits: 0, terminal integration is disabled; 1, terminal integration is enabled.

Default: 1

InterfaceName

Defines the communication interface name, that is, the UNIX socket name with which the Privileged User Password Management Agent handles requests. The socket file is located in the /opt/CA/AccessControl/data/PUPMAgent directory.

Default: PUPMAgentInterface

OperationMode

Specifies the Privileged User Password Management Agent work mode.

Limits: 0, the Privileged User Password Management Agent is disabled and not running; 1, the Privileged User Password Management Agent is enabled, running but not logging data to trace files; 2, the Privileged User Password Management Agent is enabled, running, and logging data to trace files.

Default: 0

seagent

In the [seagent] section, the tokens control the behavior of the seagent daemon.

debug_backup

Specifies whether CA Access Control uses a seagent debug messages backup file.

Limits: yes, no

Default: yes

debug_backup_file

Defines the name of the seagent debug messages backup file.

Default: *ACInstallDir*/log/seagent_debug.back

debug_file

Defines the name of the file to which CA Access Control writes seagent debug messages.

Default: *ACInstallDir*/log/seagent_debug

debug_level

Specifies the minimal level of debug messages that CA Access Control writes to the debug file.

Limits:

- disabled—no messages are written to the debug file
- critical—CRITICAL messages are written to the debug file
- very_high—CRITICAL and VERY_HIGH messages are written to the debug file
- high—CRITICAL, VERY_HIGH, and HIGH messages are written to the debug file
- normal—CRITICAL, VERY_HIGH, HIGH, and NORMAL messages are written to the debug file
- low—CRITICAL, VERY_HIGH, HIGH, NORMAL, and LOW messages are written to the debug file

Default: critical

watchdog_check_interval

Defines the time interval, in seconds, at which seagent checks that seoswd exists.

Note: This token applies only if there is a high volume of incoming connections to seagent. If seagent is idle, it checks that seoswd exists every 3 seconds and this token is ignored.

Default: 30

seauxd

In the [seauxd] section, the tokens determine the usage and refresh interval of the Unicenter TNG calendar and help manage name resolution.

client_request_timeout

Specifies the time interval, in seconds, to keep a request for resolution.

Default: 120

file_time_check

Specifies the time interval, in seconds, to check for changes in /etc/passwd.

Specifying **0** disables checking.

Default: 10

init_delay

Specifies the time, in seconds, to wait for seauxd to start up.

Default: 10

log_file_name

Specifies the name of the auxiliary log file. Its location is SEOSPATH/log.

Default: seauxd.log

log_file_size

Specifies the maximum size, in KB, of the auxiliary log file. If size is exceeded, the file is truncated to 0.

Default: 100

log_level

Specifies the level of logging to be used.

Valid values include the following:

0-Minimum info
1-ERR
2-WARN + ERR
3-NOTIC + WARN + ERR
4-DEBUG + INFO + WARN + ERR

Default: 0

req_poll_timeout

Specifies the time interval, in milliseconds, to wait for input requests.

Default: 200

respawn_seauxd_delay

Defines the minimum time in seconds at which, if seauxd quits, seosd will respawn it.

Default: 60

TNG_cal_lib

Specifies the name of the shared library containing the Unicenter TNG calendar.

Default: libcalendar

TNG_calendars

Specifies whether to use the Unicenter TNG calendar to restrict resources at set time intervals.

Default: no

TNG_lib_path

Specifies the path for CA Access Control to find the shared library containing the Unicenter TNG calendar.

Default: /opt/CA/CAlib

TNG_refresh_interval

Specifies the refresh interval, in minutes, for CA Access Control to retrieve active calendar information from Unicenter TNG.

Default: 10

trace_cnt

Indicates whether to write counters in trace file.

Valid values are yes and no.

Default: no

segrace

In the [segrace] section, the tokens determine the attributes of the segrace utility.

sepass_command

Specifies the location of the CA Access Control password replacement command that is executed when a user has no remaining grace logins.

Default: *ACInstallDir*/bin/sepass

More information:

segrace Utility—Display User Login Information (see page 158)

seini

In the [seini] section, the tokens determine the attributes of the seini intelligent search feature.

get_error_warning

Specifies whether the error and warning messages for the intelligent search feature display.

Default: yes

perform_action

Specifies whether seini performs its operations on the token or section found by the intelligent search feature.

Valid values are yes and no.

If this token is set to **yes**, the section and token, found by the additional intelligent search, are used for the requested seini operation.

Default: no

use_intelligent_search

Specifies whether to perform an intelligent search when you invoke the seini utility.

Default: no

More information:

seini Utility—Manage Configuration Files (see page 163)

selock

In the [selock] section, the tokens control the behavior of the selock utility.

unlocking_user

Specifies the name of a user, other than the owner, who can unlock a locked screen.

Default: root

More information:

selock Utility—Lock the X Terminal Screen (see page 170)

selogrd

In the [selogrd] section, the tokens control the behavior of the log routing daemons selogrd and selogrcd.

Caudit_size

Specifies the maximum size, in KB, of the audit collection file, before selogrcd creates a backup file and opens a new file.

The minimum value is 50 KB.

Default: 1024

CBackUp_Date

Sets the criterion by which selogrcd performs the backup.

Valid values include: none, yes, daily, weekly, and monthly.

If you specify **yes**, CA Access Control performs backups according to the size limit token Caudit_size and timestamps the file.

If you specify **none**, CA Access Control performs the backup according to the Caudit_size token but does not timestamp the file.

If you specify **daily**, **weekly**, or **monthly**, selogrcd adds a timestamp when it first creates the file. When the current date passes the timestamp, CA Access Control automatically creates a backup file and timestamps it.

However, if the size of the file exceeds the value of the Caudit_size token first, CA Access Control creates a backup file without issuing a timestamp.

Default: NONE

ChangeLogFactor

Specifies the factor applied to the value in the token *Interval* before testing whether the log file was changed to a backup file. For example, if the *Interval* token is set to 5 and the *ChangeLogFactor* token is set to 5 (the default), CA Access Control waits 25 seconds before checking whether the log file was changed to a backup file.

Default: 5

CipherName

Specifies the name of the file that contains the encryption functions used by selogrd if the UseEncryption token is set to eTrust.

This file must be placed in the *ACInstallDir*/lib/ directory.

The CipherName is a symbolic link to a shared object file.

Default: adcipher

CollectFile

Specifies the name of the file in which the audit collector daemon selogrcd stores the collected audit records.

Default: *ACInstallDir*/log/seos.collect.audit

CollectFileBackup

Specifies the name that selogrcd uses when backing up and renaming the file of collected audit records when it receives the USR1 signal.

Default: *ACInstallDir*/log/seos.collect.bak

ConsolePort

Specifies the name or port number for selogrd - secmon communication. It is necessary only if you plan to run both selogrcd and secmon on the same host.

If specified, seolgrd - secmon communication is done using the specified port; otherwise they use the port specified in the *ServicePort* token, or use RPC portmapper to dynamically allocate a port if that token is also empty. The service name must be a UDP port because the log routing daemon uses UDP for communication.

If the token value is a number, daemons bind to the specified port number.

If the token value is a service name (string), /etc/services or NIS services maps are used to resolve the port number.

Default: Token not set (value taken from *ServicePort* token)

DataFile

Specifies the name of the file to which the target routing information is written before being delivered to the specified targets.

Default: *ACInstallDir*/log/logroute.dat

Interval

Specifies the time interval, in seconds, between each poll of the log file by the selogrd daemon.

Default: 5

KeyFile

Specifies the name of the file that holds the audit encryption key.

This key is used when selogrd performs CA Access Control audit encryption. The location of key file is *ACInstallDir*/lib directory.

The key can be changed by sechkey utility.

Default: adcipher.bin

Mailer

Specifies the name of the program that selogrd uses to send email.

Note: This option is relevant only if you set the UseSmtpMail token to yes.

Default: /bin/mail

MaxErrorSending

Specifies whether selogrd will send error messages to syslog regarding difficulties sending audit records to selogrcd, only after the number of difficulties surpasses this token value.

The default value is 1, which means that every time selogrd has difficulties sending to selogrcd, it sends a message to syslog.

Default: 1

MaxSeqNoSleep

Specifies the maximum number of log records scanned by selogrd without sleeping.

Default: 50

RefuseUnencrypted

Specifies whether selogrcd will accept unencrypted audit. It is used in conjunction with the UseEncryption token and is redundant if UseEncryption is set to **no**. It is therefore applicable only if selogrcd uses encryption.

Valid values are:

yes- refuse unencrypted audit

no- accept both encrypted and unencrypted audit

Default: no

RouteFile

Specifies the name of the log routing configuration file. The file is used unless overridden by the selogrd utility's -config option.

Default: *ACInstallDir*/log/selogrd.cfg

SavePeriod

Specifies the time interval, in minutes, between saving information about the number of records sent.

Default: 2

sendmail_header_format

Determines the user name format in the header of mail that selogrd sends.

Note: Change this token value only if selogrd cannot send mail. (That is, if you see an error 4634 from selogrd in your syslog.)

Valid values include the following:

1-The user name format is *SmtpMailFrom*

For example: eTrust_Admin

2-The user name format is *SmtpMailFrom*@*hostnam*e (where *hostname* is the host which selogrd runs on).

For example: eTrust_Admin@machine

Default: 1

ServicePort

Specifies the name or port number that the log routing facility must use.

If specified, selogrd and selogrcd use the specified port; otherwise selogrd and selogrcd use the RPC portmapper to dynamically allocate a port.

If the token has a value, selogrd and selogrcd use the specified port; otherwise, selogrd and selogrcd dynamically allocate a UDP port using the RPC portmapper. The service name must be a UDP port because the log routing daemon uses UDP for communication.

If the token value is a number, daemons bind to the specified port number.

If the token value is a service name (string), /etc/services or NIS services maps are used to resolve the port number.

Only a UDP port/service can be specified.

Default: Token not set (selogrd and selogrcd use RPC portmapper to dynamically allocate a port)

SmtpMailFrom

Specifies the identity of the sender for UseSmtpMail.

Default: AccessControl_Admin

SmtpMailServer

Specifies the address of the remote mail server host. Use this if UseSmtpMail is set to yes. If you do not specify this token, the local machine is assumed to be the mail server.

Default: (blank - local server)

SmtpTimeLimit

Specifies the time limit, in seconds, that selogrd waits for the mail server to answer before timing out.

Default: 100

tec_conf_file

Specifies the name of the configuration file that is used for the TEC event creation by the selogrd daemon.

Default: /etc/tecad_seos.conf

UseEncryption

Determines the type of encryption.

Valid values include the following:

native-selogrd uses CA Access Control standard encryption.

eTrust-selogrd uses audit log encryption through adcipher.

no-selogrd does not use encryption.

Default: no

UseSmtpMail

Determines whether to use the direct mail feature or the previous Mailer.

Default: yes

More information:

selogrcd Daemon—Collect Audit Records (see page 269)
selogrd Daemon—Emit Audit Records (see page 270)

seos

In the [seos] section, the tokens determine the global settings that is used by CA Access Control.

admin_data

Specifies the directory where the CA Access Control Security Administrator rulers and other configuration files are stored.

Default: *ACInstallDir*/data

auth_login

Determines the login authority method. Valid values are:

native–login checks the user password against the UNIX passwd or shadow file.

eTrust–when the user does not exist in the Native environment, checks the user password against the CA Access Control database.

PAM–when the user does not exist in the Native environment, checks the login through the PAM module. This is only supported on machines where PAM is supported. PAM is used to validate the user for users such as LDAP-defined users.

Default: native

auth_module_names

Defines the language client module that is allowed to authenticate outside of native authentication. This token is set by the client inside the lca API calls before the authentication. Changing this token can affect other clients authenticating in non native mode.

No default.

fast_create_db

Specifies whether the PMDB uses the fast database copy device.

Valid values are:

no-Use the old device.

yes-Use the fast database copy device.

Default: yes

full_year

Specifies the format for displaying the year using four digits or last two digits.

For example, setting the token to yes displays 2000 instead of 00.

Valid values include the following:

yes-four digits

no-two digits

This token influences the output that is produced by secons -tv, dbmgr -d, and the seaudit utility.

Default: yes (four-digit)

ldap_base

Defines the distinguished name of the search base for user data queries in the LDAP Directory Information Tree (DIT) by CA Access Control LDAP-enabled utilities (such as sebuildla).

For example, use the following format, replacing inputs with your own:

o=*organization_name*,c=*country_name*

Default: Token not set

Important! To set up sebuildla and the required LDAP configuration settings you must to be familiar with LDAP and be able to execute the ldapsearch command. We recommend that you read the man pages for ldap(1), ldapsearch(1) and the information about setting up in the documentation for your LDAP client.

ldap_hostname

Defines a space-separated list of the host names where the LDAP servers are running for CA Access Control LDAP-enabled utilities.

Default: Token not set (localhost).

ldap_certdb_path

Defines the directory where the Netscape-style certificate database is located.

This token is required for sebuildla on platforms that use the Netscape LDAP SDK API for LDAP over SSL (Solaris). For sebuildla to work, a certificate database must contain a valid certificate for the LDAP server.

Note: sebuildla uses LDAP over SSL with server authentication (that is, no client authentication). Consult your PKI toolkit documentation for details on setting up secure services.

Default: /.netscape

ldap_keydb

Defines the name of the key database file.

Note: This setting is for AIX only as an AIX key database can have an arbitrary name (as opposed to Netscape security databases, which have names like certX.db and keyY.db depending on the implementation version, and so only the ldap_certdb_path is required for finding them).

Default: Token not set

ldap_method

Specifies the bind method that CA Access Control uses for LDAP-enabled utilities to access the LDAP service.

By default, sebuildla uses *simple* authentication with all security mechanisms. In simple authentication, ldap_userdn and the corresponding credential are passed to the LDAP server. sebuildla stores user credentials in encrypted form in ldapcred.dat at *ACInstallDir*/etc. These two parameters approximate the account and password combination that is required by the LDAP server.

Note: For SASL or TLSv.1/SSL, consult your LDAP server documentation. For a particular ldap_method setting to take effect, the corresponding mechanism must be supported and configured in the native LDAP client that is deployed on the computer where sebuildla is executed (that is, with TLS/SSL operations, valid certificates should be installed on the server and client side).

Valid values are:

0-Standard LDAP

1-SASL (RFC 2222)

2-LDAPS (LDAP over SSL - server authentication only.)

Note: The method that you use determines how you set up the ldap_userdn token and its corresponding credential (through seldapcred utility).

Default: 0

ldap_port

Defines the LDAP server port for CA Access Control LDAP-enabled utilities. Change this token if your LDAP server is not using the standard LDAP port (389).

Default: Token not set (389).

ldap_query_size

Defines the maximum number of LDAP entries sebuildla retrieves in each batch query.

Use this token when you do not want to change the LDAP server-side size limit parameter. Normally, sebuildla attempts to retrieve all data in one instance, which, if there are numerous user entries, may exceed the server's size limit and may cause the LDAP operation to fail. If you set ldap_query_size, sebuildla need not retrieve all entries for the operation not to fail. If the total number of user entries is greater than either the ldap_query_size or the server-side size limit, the number of entries that are retrieved corresponds with the lower number of these two settings.

Important! Enabling batch queries can affect sebuildla performance. Consider using this setting only where the LDAP environment has numerous user data (thousands of entries) in the DIT (Directory Information Tree).

Note: For information about server-side LDAP controls, for example, the OpenLDAP server (slapd) sizelimit parameter, consult your LDAP server documentation.

Default: Token not set (empty)

ldap_timeout

Defines the maximum amount of time (in seconds) that CA Access Control LDAP-enabled utilities wait when binding to the LDAP service and obtaining LDAP search results, before terminating the connection. The time that it takes to retrieve information from the LDAP service depends on how fast the LDAP service is, and how much user data is stored in the DIT. Use this token to account for these aspects.

Note: You may also need to adjust server-side LDAP controls to avoid truncated search results. For example, for the OpenLDAP server (slapd) you need to adjust the sizelimit parameter. Consult your LDAP server documentation for more information.

Default: Token not set (15 seconds)

ldap_uid_attr

Defines the name of the attribute that contains the user name in the LDAP DIT. RFC 2307 (An Approach for Using LDAP as a Network Information Service) prescribes *uid* as this attribute, which is the default value for this token. Change this token to let CA Access Control LDAP-enabled utilities operate against LDAP DITs with non-standard schemas.

Default: Token not set (uid).

ldap_uidNumber_attr

Defines the name of the attribute that contains the UID number in the LDAP DIT. RFC 2307 prescribes *uidNumber* as this attribute, which is the default value for this token. Change this token to let CA Access Control LDAP-enabled utilities operate against LDAP DITs with nonstandard schemas.

Default: Token not set (uidNumber).

ldap_user_class

Defines the name of the object class that contains the user data in the LDAP DIT. RFC 2307 prescribes *posixAccount* as this object class, which is the default value for this token. Change this token to let CA Access Control LDAP-enabled utilities operate against LDAP DITs with nonstandard schemas.

Default: Token not set (posixAccount).

ldap_userdn

Defines the distinguished name (DN) of the LDAP user that CA Access Control LDAP-enabled utilities use for retrieving user data from the LDAP DIT. Based on RFC 2307, CA Access Control expects to find the user data in the *uid* and *uidNumber* attributes of the *ou=People* level in the DIT. For security reasons, we recommend that this user (ldap_userdn) is given access to this data only.

If anonymous access to the DIT is permitted, you can keep this token empty. Otherwise, you must set this token and must run the seldapcred utility for CA Access Control LDAP-enabled utilities to authenticate to the LDAP service (you only need to do this once as seldapcred stores your encrypted credential in a file for reuse).

For example, set this token as follows:

```
ldap_userdn = uid=user1,ou=People,dc=myCompany,dc=com
```

Default: Token not set

ldap_userinfo_ladb

Specifies whether to retrieve user information from the LDAP Directory Information Tree (DIT).

Limits: yes, no

Default: no

ldap_verbose

Specifies whether to enable detailed account of LDAP operations involved in sebuildla getting user data.

Use this setting when you set up LDAP data retrieval in sebuildla or when troubleshooting.

Valid values are **0**-disabled; a non-zero integer-enabled.

Default: 0

locale

Determines the language for the CA Access Control daemons and utilities. CA Access Control can function in several languages.

Supported languages include: C, Japanese, Chinese-s, Chinese-t

For the complete list of languages, see /etc/ca/localeX/calocmap.txt; on Linux, see */opt/CA/SharedComponents/cawin/locale/.*

Default: C

pam_enabled

Valid on SOLARIS, HP-UX, and LINUX only.

Specifies whether the local host enables use of PAM for authentication and password changes in the LDAP database.

To do that, it checks whether the PAM library can be dynamically loaded (the library must exist on your system).

Valid values are: 'no', 'yes'.

Default: yes

parent_pmd

Defines a comma-separated list of policy model databases (PMDBs) from which this computer accepts updates. The local CA Access Control database rejects updates from any PMDB that is not specified in this list.

You can also specify a file path that contains a line-separated list of PMDBs.

Set this token to "_NO_MASTER_" for the local CA Access Control database to accept updates from any PMDB.

If you do not set this token, the local CA Access Control database does not accept updates from any PMDB.

Each PMDB is specified in the following format: pmd_name@hostname

For example:

```
parent_pmd = pmd1@host1,pmd2@host1,pmd3@host2
parent_pmd = /opt/CA/AccessControl//parent_pmdbs_file
```

Default: Token is not set (database does not accept updates from any PMDB).

Note: sepass does not support multiple destinations on the parent_pmd token.

passwd_pmd

Specifies the PMDB to which sepass sends password updates.

If you do not set this token, it inherits the value of the parent_pmd token.

The format is *pmd_name@hostname.*

The parent_pmd and passwd_pmd tokens can have the same value. If the values in the parent_pmd and passwd_pmd tokens are not the same, the passwd_pmd database sends its updates to the parent_pmd database for distribution. Therefore, the parent_pmd database must be a child (subscriber) of the passwd_pmd database.

No default.

Note: sepass does not support multiple destinations on the passwd_pmd token.

ReverseIpLookup

Controls the way seagent identifies the connecting client.

Valid values include the following:

yes-seagent looks up the IP address of the open client's socket.

no-seagent uses the host name as received from the client; seagent does not resolve any host names. (The same effect can be achieved by disabling class TERMINAL.)

Default: yes

secondary_pmd

Specifies the PMDB used as the secondary target for password replacement for users who are not defined in the primary target (passwd_pmd).

The format is *pmd_name@hostname*.

No default.

SEOSPATH

Specifies the directory in which CA Access Control is installed.

You can install CA Access Control in any directory, *if* it is not *on* an NFS-mounted file system.

Default: *ACInstallDir*

SyncUnixFilePerms

Specifies whether CA Access Control should synchronize its ACL permissions with the ACL and other permissions of the native UNIX system, if they exist.

Valid values include the following:

no-Do not synchronize the UNIX file permissions with CA Access Control ACLs.

warn-Do not synchronize ACL permissions, but issue a warning if the permissions in CA Access Control and UNIX conflict.

traditional-Change rwx permissions for the group and the owner according to CA Access Control ACLs, issue a warning in all other cases.

acl-Change native file-system ACLs according to CA Access Control ACLs (on platforms that support ACLs).

force-Functions the same as traditional or acl (on platforms that support ACLs), but also forces mapping defaccess to "other" permissions.

Note: On HP-UX and Sun Solaris 2.5 (and above), support is provided for file system ACLs. On other platforms and operating system versions, only traditional permissions mode of a file are supported.

Default: no

TNG_Environment

Specifies whether the database is created with special Unicenter TNG classes and resources.

Valid values include the following:

0-Create the database without the special Unicenter TNG classes.

1-Create the database with all the special Unicenter TNG classes.

Default: 0

TNGDir

Specifies the directory where Unicenter TNG is installed.

Valid values are the base Unicenter TNG directory (or .uniprodloc).

No default

TRUEPATH

Specifies the directory where CA Access Control is physically located. The CA Access Control directory may be a symbolic link to another physical location. This token points to the actual physical location where CA Access Control is installed.

Default: *ACInstallDir*

use_rpc_protocol

Determines whether the RPC portmapper is required. The presence of the RPC portmapper is required if you want to use the old (1.43) CA Access Control protocol. The old protocol is required to support NIS+ password changes.

This token replaces the old_protocol token.

Valid values include the following:

yes-Use the RPC portmapper to assign the port.

no-Use the port that is specified by the ServicePort token.

Default: no

More information:

sebuildla Utility—Create a Lookaside Database (see page 104)
seldapcred Utility—Encrypt and Store a Credential (see page 168)
sepass Utility—Set or Replace a Password (see page 182)

SEOS_syscall

In the [SEOS_syscall] section, the tokens are used by the SEOS_syscall kernel module.

bypass_NFS

Determines whether to bypass NFS files from SEOS events.

Valid values include the following:

0-Do not bypass NFS files.
1-Bypass NFS files.

Default: 0

bypass_realpath

Specifies whether to bypass real file paths resolution for authorization.

If you enable this setting (1), CA Access Control does not resolve file paths for authorization. This accelerates file events handling However, generic rules will not be enforced for file accesses that are made using links.

Example: A deny access rule for /realpath/files/* is not considered if this setting is enabled and a user accesses a file in this directory from a link. You need to have a generic rule for the link too (/alternatepath/*).

Default: 0

cache_enabled

Determines whether to use caching for full path resolution to determine access permissions for files.

Valid values include the following:

0-No caching.
1-Use caching.

Default: 0

cache_rate

Determines the cache rate that used when cache is enabled for full path resolution.

Bigger values mean better caching.

Default: 10000

call_tripAccept_from_seload

Determines whether to call tripAccept from the seload command after CA Access Control starts and, if tripAccept is called, defines a list of comma-separated TCP/IP ports that tripAccept should connect to and wake up the ports' listeners.

Valid values are any TCP/IP port number, and:

0-Do not call tripAccept from seload.

Limits: 0-64000

Default: 0

cdserver_conn_res

Determines whether to treat T_CONN_RES streams messages as high priority messages in the fiwput routine on UnixWare.

Valid values are:

1-handle T_CONN_RES streams messages as high priority messages in the fiwput routine.

0-handle T_CONN_RES streams messages as low priority messages in the fiwput routine.

Default: 0 (on UnixWare it should be 1)

debug_protect

Determines whether to allow debugging of any program while CA Access Control is running.

Valid values include the following:

0-Debugging allowed.
1-Debugging not allowed.

Default: 1

DESCENDENT_dependent

Determines whether a descendent of a SEOS daemon can register a SEOS service.

Valid values include the following:

0-Anyone can register a SEOS service.
1-Only a descendent can register a SEOS service.

Default: 0

exec_read_enabled

Specifies whether the CA Access Control kernel identifies script execution.

Valid values include the following:

0-CA Access Control kernel does not identify script execution.

1-CA Access Control kernel identifies script execution.

Default: 0

Note: If the Privileged User Password Management Agent is installed on the endpoint, the default value is 1. When enabled, the Privileged User Password Management Agent is able to identify shell scripts named that use the Privileged User Password Management Agent file (acpwd) without defining the script as a PROGRAM resource.

file_bypass

Indicates whether CA Access Control checks file access for files that are not defined in the database. By default CA Access Control does not check files that are not defined in the database.

Valid values include the following:

-1-Do not check all files.
0-Check all files.

Default: -1

GAC_root

Determines whether to use GAC caching for files when the user is root. By default GAC is not used when the user is root.

Valid values include the following:

0-No caching for root user.
1-Use caching for root.

Default: 0

HPUX11_SeOS_Syscall_number

Determines the default syscall number to communicate with SEOS_syscall on HP-UX.

Valid values include any unused syscall entry number in sysent.

Default: 254

kill_signal_mask

Defines which signals to protect.

Valid values include a mask that ORs (includes) all the signals that we want SEOS events for.

Default: SIGKILL, SIGSTOP, or SIGTERM events. Actual value varies by platform:

- HP-UX: 0x804100
- Sun Solaris: 0x404100
- IBM AIX and Digital DEC UNIX: 0x14100
- Linux: 0x44100

link_protect

Determines whether a symbolic link will be protected.

Valid values include the following:

0-Links are not protected.
1-Links are protected.

Default: 0

max_generic_file_rules

Defines the maximum number of generic file rules allowed in the database.

Note: A large number may cause strange behaviors on different platforms. For assistance, contact CA Support at http://ca.com/support.

Valid values include any number greater than (<) 511.

Note: This token is supported only on AIX, HP, Linux, and Solaris.

Default: 512

max_regular_file_rules

Defines the maximum number of file rules allowed in the database.

Note: A large number may cause strange behaviors on different platforms. For assistance, contact CA Support at http://ca.com/support.

Valid values include any number greater than (<) 4095.

Note: This token is supported only on AIX, HP, Linux, and Solaris.

Default: 4096

mount_protect

Determines whether to allow mount and unmount of directories used by CA Access Control.

Valid values include the following:

0-Allow mounting.
1-Do not allow mounting.

Default: 1

proc_bypass

Determines whether to check file access when a file belongs to a process file system (/proc). Valid values include the following:

0-token is ignored

1-bypass file access checks

Default: 1

SEOS_network_intercept_type

Specifies the type of network interception to use (HP-UX only).

Note: You must also set SEOS_use_streams = yes

Valid values are:

0 - TCP hook

1 - streams

Default: 1

Important! Do not modify this token yourself. For assistance, contact CA Support at http://ca.com/support.

SEOS_streams_attach

Specifies whether CA Access Control attaches to running STREAMS.

If you change this setting, you need to restart daemons that already listen to the network for CA Access Control to protect them.

Note: This setting applies only to Solaris 9 or earlier.

Default: yes

SEOS_unload_enabled

Determines whether the SEOS_syscall kernel module can be unloaded.

Valid values include the following:

0-Do not allow the unload.
1-Allow the unload.

Default: 1

SEOS_use_ioctl

Specifies the CA Access Control kernel module communication method (ioctl or system call).

You can use the *ioctl* communication method when all available system call numbers are in use by the operating system.

Values: 0-system call **1**-ioctl

Default: 0

Important! Do not modify this token yourself. For assistance, contact CA Support at http://ca.com/support.

SEOS_use_streams

Specifies whether to use the streams subsystem for network interception (whether SEOS_load automatically pushes a module into streams).

This settings can only be used for HP-UX and Sun Solaris versions 8 and 9.

Default: no

silent_admin

Defines the user IDs of the maintenance users. This user's activity is permitted when security is down and silent_deny is *yes*. Use the user's numeric UNIX UID to define the maintenance user.

Default: 0 (user ID of root)

silent_deny

Determines whether to deny any event when security is down.

Valid values include the following:

yes-Silent deny is enabled (maintenance mode).

no-Silent deny is disabled.

Default: no

STAT_intercept

Specifies whether to check file access when a stat system call occurs.

If you specify 1 (check file access), CA Access Control does not let users who do not have *read* permissions perform operations that get information about a file and records *read* in the audit log. If you set this to 0, any user can get file information.

Values: 0 (do not check file access), 1 (check file access).

Default: 0

STOP_enabled

Determines whether to use the STOP feature, which protects from stack overflow attacks.

Valid values include the following:

0-Off.
1-On.

Default: 0

synchronize_fork

Determines how fork synchronization is managed.

On *HP-UX platforms*

1-Report forks from parent
2-Report forks from child

On *other platforms*

1-Parent reports without synchronization
2-Parent reports with synchronization (not supported on Linux)

Limits: Any value lower than 1 is interpreted as 1. Any value greater than 1 is interpreted as 2.

Note: Do not modify this setting because it may cause strange behaviors on different platforms. For assistance, contact CA Support at http://ca.com/support.

Default: 1

syscall_monitor_enabled

Specifies whether CA Access Control monitors processes that are executing CA Access Control code. If you have this enabled (the default), you can use the *secons -sc* or *secons -scl* to view these processes.

Valid values are:

0-inactive

1-active

Default: 1

threshold_time

Defines how long, in seconds, an intercepted system call can be blocked before it is considered risky. If a process is blocked for a period that is longer than this time, CA Access Control reports that SEOS_syscall module unload may fail.

Note: This value affects the unload readiness reports CA Access Control provides. For more information, see the *Enterprise Administration Guide*.

Default: 60

trace_enabled

Determines whether to use the SEOS_syscall circular trace buffer.

Valid values include the following:

0-Do not use tracing.
1-Use tracing.

Default: 0

use_tripAccept

Determines whether to use the tripAccept utility when unloading SEOS_syscall to wake up the blocked accept system calls. This avoids running SEOS_syscall code after the module is unloaded.

Valid values are yes and no.

Default: yes

seosd

In the [seosd] section, the tokens determine the behavior of the authorization daemon and the cache utility for performance improvement.

bypass_filenames

Specifies a file that contains a list of file names to be exempted from seos events.

For example, bypass_filenames = /opt/CA/AccessControl//bin/bypass_filenames

Default: Token not set

bypass_nfs_port

Specifies whether the port used by nfs (port 2049) are bypassed for CONNECT. The bypass exists to let NFS function correctly.

If you change the value of this token to *no*, there will be no bypass for this port. Make sure that you then provide the required CA Access Control rules to replace this bypass. Following is an example of such rules (you *cannot* use them as is):

```
nr hostnet all mask (0.0.0.0) match(0.0.0.0)
nr TCP 2049 owner(nobody) defaccess(none)
authorize TCP 2049 hostnet(all) access(w) uid(root)
nr TCP nfsd owner(nobody) defaccess(none)
authorize TCP nfsd hostnet(all) access(w) uid(root)
```

Note: If you set the value of this token to *no* but do not provide the correct CA Access Control rules, NFS stops working.

Default: yes

bypass_outgoing_TCPIP

Defines a comma-separated list of ports for which seos_syscall will not pass outgoing connection events to seosd.

Default: Token not set

bypass_suid_for_login

Specifies the path of the login program for which the dummy SUID system calls should be ignored.

This is used in case of some login programs, such as samba, which generate a large number of dummy SUID system calls. These system calls may interfere with the correct recognition of the logging in user.

Default: none

bypass_suid_program

Allows multiple su commands. On some platforms, the system's su binary works in a nonstandard way: When an su command to a non-root user is requested, it executes su to root prior to executing su to the requested user.

If CA Access Control surrogate protection is set for the root user, it may prevent the successful execution of an su to non-root users as well.

To use the surrogate protection for the root user on such platforms and still to be able to su to non-root users without interruption, set the bypass_suid_program token to contain the real path for the system's su binary.

Default: none

bypass_system_files

Determines whether the CA Access Control authorization engine should bypass read access for the /etc/passwd and /etc/group system files.

Valid values are:

yes-bypasses read access to system files.

no-does not bypass read access to system files.

Default: yes

bypass_TCPIP

Allows you to add one or more ports separated by commas for which seos_syscall will not pass events to seosd.

The syntax is bypass_TCPIP=*port1*[,*port2*,*portx*]

Default: Token not set

bypass_xdm_ports

Specifies whether the ports used by xdm (ports 6000-6010) are bypassed for CONNECT. The bypass exists to let xdm function correctly.

If you change the value of this token to *no*, there will be no bypass for these ports. Make sure that you then provide the required CA Access Control rules to replace this bypass. Following is an example of such rules (you *cannot* use them as is):

```
nr hostnet all mask (0.0.0.0) match(0.0.0.0)
nr TCP X-Win owner(nobody) defaccess(none)
authorize TCP X_Win hostnet(all) access(r)
authorize TCP X_Win hostnet(all) access(w) uid(root)
authorize TCP X_Win hostnet(all) access(w) gid(mygroup)
nr TCP 6000 owner(nobody) defaccess(none)
authorize TCP 6000 hostnet(all) access(r)
authorize TCP 6000 hostnet(all) access(w) uid(root)
authorize TCP 6000 hostnet(all) access(w) gid(mygroup)
```

Note: If you set the value of this token to *no* but do not provide the correct CA Access Control rules, xdm stops working. If the value of this token to *yes* and an outgoing connection is made via ports 6000-6010, the class name in the corresponding audit record is TERMINAL.

Default: yes

cron_program

Improves the check for cron login in seosd.

Set the cron_program token to contain the real path for the system's cron binary.

Default: none

dbdir

Specifies the location of the CA Access Control database.

Default: *ACInstallDir*/seosdb

device_file

Specifies whether to scan all devices in /dev.

When the value of this token is set to Yes and the tty is not found in the standard list, CA Access Control scans all the devices located in /dev.

(qplib resolves the tty name from the standard devices.)

Note: You can add devices to the list of the tty names.

Default: no

dns_server

Specifies the DNS server name used to change host resolving from the default server to another server.

This token is usually used when the DNS caching option is enabled.

Default: none

domain_names

Specifies a list of domain names that seosd appends to short host names it receives for authorization purposes in order to create a fully qualified name, so that these names can be authorized in the relevant HOST, CONNECT, or TERMINAL classes.

To identify a full name, seosd tries to append domain names in the domain_names list to the short name for authorization purposes.

seosd first looks for a relevant rule in its database, using the short name only. If it does not find a record that matches the short name, it appends each domain name specified in the domain_names token, one by one, until it finds a match.

For example, suppose you assign domain_names the following list:

domain_names= market.com, journey.com, total.com

Here is how seosd handles the matching process when a request from a subscriber called *acme*-which was not defined as a rule in the database-comes in:

acme (not found in database)
acme.market.com (not found)
acme.journey.com (not found)
acme.total.com (found)

seosd uses the first record that matches (acme.total.com in this example) for authorization purposes.

Default: As defined in /etc/resolv.conf

EnablePolicyCache

Determines whether a run-time table should be used to store the database values required for authorization. The run-time table is loaded to the memory when seosd starts. This avoids connecting to the database and thus reduces the authorization time.

Valid values are yes and no.

Default: no

enf_register

Determines whether seosd registers to Unicenter NSM Event Notification Facility (ENF).

The valid values include the following:

yes-seosd registers to the ENF.

no-seosd does not register to the ENF.

Default: no

FileCache_auths

If caching is enabled, specifies the number of records in the authorization pool. The maximum number of authorization records that can be cached is 800.

Default: 80

FileCache_CleanInt

Specifies how often to erase the file cache (in minutes).

Default: 60

FileCache_files

If caching is enabled, specifies the number of records in the file pool. The maximum number of file records that can be cached is 200.

Default: 20

FileCache_InitPrio

Specifies the initial priority value of new records in the cache table.

Default: 10

FileCache_PriorInt

If caching is enabled, specifies the frequency of recalculating priorities in the cache table. Each time a new record is saved counts as one.

Default: 1

FileCache_users

If caching is enabled, specifies the number of records in the user pool. The maximum number of user records that can be cached is 500.

Default: 50

get_login_terminal

Determines whether seosd attempts to find the peer address of the login program in an alternative way. This is useful for connections such as ssh.

Valid values include yes and no.

Default: yes

grace_admin

Determines the number of the grace logins that are set when an administrator changes users' passwords.

Default: Token not set (1)

GroupidResolution

Determines how CA Access Control resolves GID numbers to group names.

Valid values include the following:

system-CA Access Control uses a system call to translate gid numbers. This value can be used for stand-alone, DNS client, and DNS server stations. (See also the resolve_timeout token in this table.)

cache-gid numbers and group names are cached in seosd. This is the fastest and easiest way to do translations but the cache cannot be updated during runtime.

ladb-CA Access Control uses a lookaside database to translate gid numbers. The sebuildla utility must be run to recreate the lookaside database each time an update to the relevant transaction table takes place.

For NIS, and NIS+ servers, you can use either cache or ladb.

For Sun Solaris 2.5 and above and HP-UX 11.x, you can use either cache or ladb.

For all stations, the value ladb is preferred.

Default: Token not set (system)

HostResolution

Determines how CA Access Control resolves IP addresses to host names.

Valid values include the following:

system-CA Access Control uses a system call to translate IP addresses. This value can be used for stand-alone, NIS/NIS+ client, and DNS client stations. (See also the resolve_timeout token in this table.)

cache-Host names and their IP addresses are cached in seosd. This is the fastest and easiest way to do translations but the cache cannot be updated during runtime.

ladb-CA Access Control uses a lookaside database to translate IP addresses. The sebuildla utility must be run to recreate the lookaside database each time an update to the relevant transaction table takes place.

For NIS, NIS+, and DNS servers, you can use either cache or ladb; the value ladb is preferred.

Default: Token not set (system)

IsolatedDaemon

Determines whether seosd closes the file descriptors stdin, stdout, and stderr when they become a daemon.

Valid values include the following:

yes-seosd closes these file descriptors when they become a daemon.

no-seosd does not close these file descriptors when they become a daemon.

Default: no

kill_ignore

Specifies whether seosd ignores (denies) the “kill -9” command directed toward any one of the three main CA Access Control daemons. Valid values include the following:

yes-Ignores the kill command. This is the default value.

no-The kill command terminates seosd.

Default: yes

login_parent_check

Specifies whether the parent process should continue (once a child process has logged in) with the login sequence or abandon the sequence and inherit the login from the child.

Valid values are 0 and 1.

If it is 0, the parent continues with the login sequence.

If it is 1, the parent abandons the login sequence and inherits the login from the child.

Default: Token not set (0)

lookaside_allowdupuid

Determines whether sebuildla will register duplicate UIDs

Valid values:

yes-register duplicate UIDs

no-in case of duplicate UIDs, register only one UID

Note: Duplicate UIDs may cause inconstancy On UNIX OS

Default: no

lookaside_path

Specifies the directory where the lookaside database is located. Create this directory before running the sebuildla utility.

Note: The lookaside database files are built and updated using the sebuildla utility.

Default: *ACInstallDir*/ladb

max_loggedin_users

Defines the maxinum number of logged in users.

Note: This value determines the size of one of the internal memory tables. The larger the table, the more memory it consumes.

Limits: 4096-20480

Default: 8192

MultiLoginPgm

Defines the name and full path of a program that performs multiple logins. It is used to detect the correct login sequence for these special login applications.

MultiLoginPgm is the login application name with the full path.

Default: none

network_cache_timeout

Specifies the time interval, in minutes, between network cache-table cleanings, if network cache is used. Use this token to set time limits for the stored accepted incoming TCP requests.

Note: For more information about using the network cache, see the *Endpoint Administration Guide for UNIX*.

Default: 10

nfs_devices

Specifies the name and path of the file that contains the NFS major device numbers. Specify the full file path.

CA Access Control uses this file if it fails to get the program using device and inode number and also fails to get it using its name. The file contains the NFS defaults for major device numbers for every platform. This may vary from system to system. To find the numbers for your system, use a small program with the UNIX getmajor() function. Then, edit the nfsdevs.init file (or the file you named with this token) to contain the numbers you find.

Note: Whenever you mount and remount the NFS system, you should update your nfsdevs.init file. You can also use the first four digits of the device only. These numbers remain unchanged, even when you unmount and remount the system.

Default: *ACInstallDir*/etc/nfsdevs.init

protect_bin

Specifies whether seosd protects the CA Access Control binary files. Specify one of the following values:

yes-seosd protects the CA Access Control binary files unless rules that allow such access are defined.

Note: Do not specify yes when the _default access for your FILE records is none because, unless all /opt/CA/AccessControl//bin files have FILE records, inaccessibility of files could make CA Access Control unusable.

no-seosd does not protect the CA Access Control binary files.

Default: no

resolve_rebind

Specifies if seosd re-establishes the connection to the NIS server after a time-out failure.

We strongly recommend that you do not change the default value.

Default: yes

resolve_timeout

Specifies the maximum number of seconds seosd tries to resolve IP to address, user ID to user name, group ID to group name, or service port number to service name.

The value takes effect in two cases:

When seosd is using system resolution. (See the HostResolution, ServiceResolution, UseridResolution, and GroupidResolution tokens.)

When the under_NIS_server token is set to no.

If the specified time expires without a resolution, seosd assumes that no resolution exists for the specified IP, ID, or port.

If value is set to 0, there is no time out.

Default: 5

rt_priority

Determines whether seosd has real-time priority.

Valid values are yes and no

When this token is set to yes, seosd will have real-time priority.

Default: yes

ServiceResolution

Determines how CA Access Control translates TCP port numbers to service names.

Valid values include the following:

system-CA Access Control uses a system call to translate TCP port numbers. This value can be used for stand-alone, NIS/NIS+ client, DNS client, and DNS server stations. (See also the resolve_timeout token in this table.)

cache-Service names and their TCP port numbers are cached in seosd. This is the fastest and easiest way to do translations but the cache cannot be updated during runtime.

ladb-CA Access Control uses a lookaside database to translate TCP port numbers. The sebuildla utility must be run to recreate the lookaside database each time an update to the relevant transaction table takes place.

For NIS, and NIS+ servers, use either cache or ladb.

Default: system

sim_login_timeout

Defines the timeout (in minutes) before CA Access Control removes unused simulated login user entries from the Accessor Element Entry table (ACEE).

CA Access Control performs a simulated login to create ACEE entries when it needs access to information that can be found in the ACEE.

Default: 60

special_check

Specifies whether to enable file path checking on kernel module loading. When enabled, CA Access Control checks that the kernel module to be loaded matches the filepath property of the KMODULE record (for non-Linux systems), or matches the signature of the KMODULE record (for Linux systems).

Default: no

terminal_default_ignore

Determines whether the defaccess value of the _default TERMINAL and of the specific TERMINAL records are considered when authorizing administrative access.

Valid values are yes and no.

yes-Administrative access ignores the defaccess value of the _default and of any specific TERMINAL records. In this case, administrative access will require an explicit authorization rule for a relevant specific TERMINAL record.

no- Administrative access considers the defaccess value of all relevant TERMINAL records whether it is _default or specific.

Default: yes

terminal_search_order

Specifies whether seosd tries to check a TERMINAL defined by name before trying it by its IP address.

Valid values are:

name - TERMINALs will be checked by name before IP address.

ip - TERMINALs will be checked by IP address before name.

Note: TERMINAL class supports generic rules defined by wildcards (IP address or host name pattern match). Generic rules are *always* checked after specific (full-name) rules. For example, if you set this to *ip*, seosd looks for a TERMINAL resource in the following order: complete IP address match, complete host name match, IP address pattern match, host name pattern match.

Default: name

trace_file

Specifies the name of the file to which the trace messages are sent, if trace messages are requested.

Default: *ACInstallDir*/log/seosd.trace

trace_file_type

Determines whether the trace file is written in binary or text format.

Valid values include the following:

binary-The trace file should be written in binary format. This option reduces the space occupied by this file.

text-The trace file should be written in text format.

The daemon seosd checks the value of this token and compares it to the contents of the trace file. If the token value does not match the format of the trace file, seosd saves the trace file under its name and adds the extension .backup.

Default: text

trace_filter

Specifies the name and path of the file that contains the filter data that is used to filter the trace messages.

Default: *ACInstallDir*/data/*language*/etc/trcfilter.init

trace_space_saver

Specifies the amount of free space, in MB, to be left in the file system. When the amount of free space is less than this number, CA Access Control disables the trace.

Note: Trace is never automatically enabled, even if more space becomes available at a later time.

Default: 512

trace_to

Specifies the destination of trace messages.

Valid values include the following:

file-CA Access Control sends the trace messages to the file specified by the trace_file token. To disable tracing, use the *secons -t-* command. For more information, see the trace_file token in this table.

file,stop-CA Access Control generates trace messages during daemon initialization. Once the daemon is initialized, trace messages generation stops.

none-CA Access Control does not issue trace messages. This is the normal setting after you install and implement CA Access Control.

Note: If the token is set to **file** or **file,stop**, the CA Access Control trace can be toggled with the secons command with the -t option.

Default: file, stop

UpdSurrogLogin

Specifies whether CA Access Control updates the user's last access time on a surrogate login.

Valid values are:

1 - CA Access Control updates the user's last access time on a surrogate login.

0 - CA Access Control does *not* update the user's last access time on a surrogate login

Undef_ForPacl

Determines whether seosd checks an undefined user when there is an asterisk (*) in the accessor's name in a PACL.

Valid values include the following:

1-seosd will not include undefined users with an asterisk in their PACL.

0-seosd will include undefined users with an asterisk in their PACL.

Default: 0

under_NIS_server

Determines whether seosd uses internal name resolution instead of system name resolution.

Valid values include the following:

yes-seosd stores in memory or in a lookaside database (see the use_lookaside token) all user, group, host, and port information during startup.

This is required for NIS, NIS+, and DNS server machines, and for the following operating systems: Sun Solaris 2.5 and above, HP-UX 11.x, IBM AIX 4.3.x, and IRIX 6.5.

Important! Turning this token off could hang the machine if it is an NIS server or one of the previously-mentioned operating systems.

no-seosd uses system name resolution and the resolve_timeout token takes effect.

Note: This token is automatically assigned a value during installation.

This token remains for purposes of backward compatibility only. If you have a new CA Access Control installation or an installation of version 2 or higher, use the tokens HostResolution, ServiceResolution, UseridResolution, and GroupidResolution instead.

Default: Assigned during installation

use_lookaside

Determines whether seosd stores the user, group, host, and port information in a lookaside database or in memory.

Note: This token is used in conjunction with the under_NIS_server token and has no relevance unless the under_NIS_server token is set to yes.

Valid values include the following:

yes-seosd uses the lookaside database for user, group, host, and service details. The lookaside database is built by the sebuildla utility and can be refreshed by it at any time.

The location of the lookaside database is set by the lookaside_path token.

no-seosd caches all user, group, host, and service information during startup so that all translations can be done in memory. We recommend that seosd be restarted daily to refresh the cache.

This token remains for purposes of backward compatibility only. If you have a new CA Access Control installation or an installation of version 2 or higher, use the tokens HostResolution, ServiceResolution, UseridResolution, and GroupidResolution instead.

Default: no

use_mapped_user_name

(Valid if both CA Access Control and UNIX Authentication Broker are installed) Specifies whether seosd uses the user enterprise name in audit records.

Values: yes, no

Default: no

use_nfs_devices

Determines whether to use NFS devices. Valid values are yes or no.

Default: Yes

use_standard_functions

Determines whether sebuildla in an NIS environment will retrieve users by calling the standard system function getpwent or by parsing the output of ypcat passwd and cat /etc/passwd commands.

Valid values are:

yes-use the standard system function getpwent

no-use parsing of the output of ypcat passwd and cat /etc/passwd commands.

Default: yes

use_trusted_script

Specifies whether seosd will use the trusted script mechanism.

When the trusted script mechanism is used, programs called from within a shell script retain the name of the shell script in the internal CA Access Control tables.

This means that if a script was used in a PACL, these programs will inherit that privilege. This also means that you cannot protect these programs via CA Access Control.

A trusted script begins with #! on the first line.

When the trusted script mechanism is **not** used, these programs will be registered in the internal CA Access Control tables under their own names.

Default: yes

use_unab_db

(Valid if both CA Access Control and UNIX Authentication Broker are installed) Specifies whether seosd uses the UNIX Authentication Broker database to resolve users and groups name if the current method is unable to do so. This token coincides with the tokens: use_lookaside, UseridResolution, GroupidResolution.

Values:yes, no

Default: no

UseFileCache

Specifies whether to use the cache tool for file records to improve performance.

Default: yes

UseNetworkCache

Determines whether CA Access Control caches accepted incoming TCP requests.

Note: For more information about using the network cache, see the *Endpoint Administration Guide for UNIX.*

Valid values are yes and no.

Default: no

UseridResolution

Specifies how CA Access Control translates UID numbers to user names.

Valid values include the following:

system-CA Access Control uses a system call to translate uid numbers. This value can be used for stand-alone, NIS/NIS+ client, DNS client, and DNS server stations.

cache-User names and their uid numbers are cached in seosd. This is the fastest and easiest way to do translations but the cache cannot be updated during runtime.

ladb-CA Access Control uses a lookaside database to translate uid numbers. The sebuildla utility must be run to recreate the lookaside database each time an update to the relevant transaction table takes place.

For NIS and NIS+ servers, Sun Solaris 2.5 and above, or HP-UX 11.x operating systems, you must use either cache or ladb.

Default: system

watchdog_refresh

Determines whether seosd refreshes the Watchdog to scan the privileged programs and secured files for each file handle.

Valid values include the following:

yes-seosd refreshes the Watchdog.

no-seosd does not refresh the Watchdog.

Default: no

seosdb

In the [seosdb] section, the tokens manage database checking and rebuilding.

CheckAlways

Determines whether the database should be checked for corruption at CA Access Control initialization.

Valid values are yes and no.

Default: yes

CheckProgram

Specifies the full path and parameters of an alternative command to be used instead of the internal code for checking the database. The command should return 0 if the database is valid or a nonzero number if it should be corrected.

Default: Token not set (do not run any program; same as using *dbmgr -u -fast*)

CreateNewClasses

Specifies whether you can add new classes, created with the seclassadm utility, to a database.

Valid values are yes and no.

Default: yes

CreateNewProps

Specifies whether to save data about the new properties in a file when the CA Access Control sepropadm utility creates new database property.

Valid values are yes and no.

If it is yes, sepropadm saves the data about new properties in a file and when dbmgr -c utility later generates the new CA Access Control database, dbmgr uses this file to add these properties to the database.

Default: yes

RebuildAlways

Indicates whether the CA Access Control database should always be rebuilt at CA Access Control initialization.

Valid values are yes and no.

Default: no

RebuildProgram

Specifies the full path and parameters of an alternative command to be used instead of the internal code for correcting the database.

Default: Token not set (do not run any program; same as using *dbmgr -u -build all*)

seoswd

In the [seoswd] section, the tokens determine the behavior of the Watchdog.

agent_manager_check_enabled

Specifies whether to protect the AgentManager daemon.

Default Value: no

agent_manager_refresh_interval

Specifies the interval when the watchdog checks if Agent Manager daemon is running or not.

Default Value: 10 Minutes

BlockingInterval

Specifies the interval, in seconds, that the watchdog waits for a response from the main daemon. When elapsed, the watchdog sends a signal to the main daemon.

Default: 60

IgnoreScanInterval

Specifies whether to scan programs and files at specific intervals.

If the token value is no, then the watchdog performs interval scanning; if yes, then it does not scan at intervals.

Note: If you do not specify the scan times with the PgmTestTime or SecFileTestTime tokens, and this token is set to yes, then the watchdog does not scan trusted programs or secured files respectively.

Default: no

PgmRest

Specifies the period, in seconds, after the last event and before checking programs again. The program rests to prevent system overload.

Default: 10

PgmTestInterval

Specifies the time interval, in seconds, between the rescanning of trusted programs.

Note: If the value equals to or, is greater than one day (86400 seconds) then IgnoreScanInterval defaults to *yes*.

Default: 18000 (five hours)

PgmTestStartTime

Specifies the start time, in *hh*:*mm* format, of the first trusted program scan.

If you do not set this token, the Watchdog performs the first scan shortly after startup.

No default.

PgmTestTime

Specifies fixed scan times, in *hh:mm* format, for trusted programs. You can specify more than one scan time by separating them with spaces.

Note: If you do not specify scan times, and you set the IgnoreScanInterval token to yes, then the Watchdog does not scan trusted programs.

No default.

policyfetcher_refresh_interval

Specifies the interval, in seconds, to verify that the policyfetcher daemon is running.

Default: 600

ProcRestartHours

Specifies the hours when the watchdog restarts high memory size process.

Valid values: 0 - 23 (value in hours)

Default Value: 0 - 5

ProcVSizeCritical

Specifies the process critical memory size, the watchdog restarts the process immediately.

Default Value: 500 (value in megabytes)

ProcVSizeHigh

Specifies the process memory size high watermark, the watchdog restarts during the restart hours.

Default Value: 300 (value in megabytes)

ProcVSizeInterval

Specifies the interval between process memory size verification. The watchdog checks seosd and uxauthd processes.

Default Value: 900 (value in seconds)

RefreshParams

Specifies the time interval, in seconds, between successive reads by the Watchdog of the seos.ini tokens.

Default: 86400 (one day)

SecFileRest

Specifies the period, in seconds, after the last event and before checking secured files again. The Watchdog rests in order to prevent system overload.

Note: If you do not specify scan times, and you set the IgnoreScanInterval token to yes, then seoswd does not scan secured files.

Default: 10

SecFileTestInterval

Specifies the time interval, in seconds, between the rescanning of secured files.

Default: 36000 (ten hours)

SecFileTestStartTime

Specifies the start time, in *hh*:*mm* format, of the first scan of secured files.

If no value is given, the Watchdog performs the first scan a short time after CA Access Control daemons start.

No default.

SecFileTestTime

Specifies fixed scan times, in *hh:mm* format, for secured files. You can specify more than one scan time by separating them with spaces.

No default.

SeosAYT

Specifies the time interval, in seconds, between Watchdog checks of the daemon seosd.

Important! Do not modify this token by yourself because incorrect value may cause major problems in CA Access Control operation. For assistance, contact CA Support at http://ca.com/support.

Default: 60

SignalMinInterval

Specifies the interval, in seconds, between scans after a HUP signal triggers a one-time scan on demand, to protect the system against overload.

Note: Scan on demand is performed both on trusted programs and secured files.

Default: 60

UnTrustMissing

Determines whether the Watchdog should attempt to untrust a program or file, even though it cannot find it. For example, if the file was deleted or the relevant NFS partition is not mounted.

The following list includes the valid values:

yes-Attempt to untrust the missing file.

no-Do not attempt to untrust the missing file.

Default: yes

unab_check_enabled

Specifies whether to protect the authentication daemon.

Values: yes, no

Default: no

unab_refresh_interval

Specifies the interval, in seconds, to verify that the authentication daemon is running.

Default: 600

VerifyCtime

Specifies whether CA Access Control Watchdog checks the time of the last file status change of trusted programs and secure files.

Valid values are yes or no.

Default: no

serevu

In the [serevu] section, the tokens determine the attributes of the serevu utility.

config_file

Specifies the location of the serevu configuration file.

Default: *ACInstallDir*/etc/serevu.cfg

def_diff_time

Specifies the time interval during which serevu scans the relevant system log for failed logins.

The value can be specified in seconds (that is, 300) or minutes (that is, 5m).

For example, if the token is set to 300, serevu searches for failed logins that occurred during the previous 300 seconds.

We recommend that this value be an even multiple of the value in the def_sleep_time token.

Default: 5m (5 minutes)

def_disable_time

Specifies the time that a user account is disabled because of too many failed login attempts.

The value can be specified in seconds (that is, 300) or minutes (that is, 5m). You can also use the *FOREVER* value to disable user logins forever.

Important: Use the *FOREVER* value to disable user logins permanently.

Default: 6m (6 minutes)

def_fail_count

Specifies the number of failed logins each user is entitled to, per period, in the token def_diff_time.

Users with at least this number of failed logins over the specified time period are disabled.

Note: We recommend that the number of failed logins always be the same as the value of allowed unsuccessful login attempts set on your system. For example, on Sun Solaris use the RETRIES token in the /etc/default/login file to set the system value.

Default values are five for Solaris and three for HP-UX and AIX. See your operating system documentation for more details.

Default: 5

def_sleep_time

Specifies the time between successive serevu checks.

The value can be specified in seconds (that is, 120) or minutes (that is, 2m).

Default: 2m (2 minutes)

save_disable_path

Specifies the location of the disabled user accounts list so serevu can handle disabled users when it goes down.

Default: *ACInstallDir*/log/serevu_disable.users

More information:

serevu Utility—Handle Unsuccessful Login Attempts (see page 208)

sesu

In the [sesu] section, the tokens control logging on as a user other than yourself, without having to enter the password of the other user.

AlwaysTargetShell

Determines whether to use the target shell (SysV style) or the invoker shell (BSD style). If yes, CA Access Control uses the target user shell.

Valid values are yes and no.

Default: no

FilterEnv

Specifies a list of environment variables that sesu does not pass to the shell when the target user is root. Separate variable names with spaces or tabs.

No default.

old_sesu

Determines whether the old or new sesu utility is used.

Valid values include the following:

yes-Use the old sesu utility as it was in previous versions.

no-The new sesu utility calls the native su program (as defined in the SystemSu token) to ensure consistency between su and sesu. If the SystemSu token is not valid, sesu reverts to the old mechanism.

Note: If this token is set to no, the tokens Path, AlwaysTargetShell, sys_env_file, and FilterEnv are ignored.

Default: yes

Path

Specifies the value that sesu uses to set the PATH environment variable. If the token is not set, sesu does not set the PATH variable.

No default.

request_target_password

Specifies whether to request the password of the target user when the *old_sesu* token is set to no and the user is executing sesu for a non-root user.

Default: yes

sys_env_file

Specifies an ASCII file containing environment variable values for the sesu session. This token is relevant only when starting sesu with the "-" parameter (sesu -). The format for each line of the file is *variable* = *value*.

Default: None (except for IBM AIX where it is */etc/environment*)

SystemSu

Specifies the location of the /bin/su program. Update this token if you use a program in a location other than the default location. When sesu cannot find the authorization daemon, it executes the program specified in this token.

Note: On AIX, replace the system su binary with a symbolic link to the sesu wrapper instead of the sesu binary.

Default: /bin/su

UseInvokerPassword

Determines whether sesu requires the invokers to specify their own passwords. If the token value is no, sesu does not require any password.

Default: no

More information:

sesu Utility—Substitute User (see page 212)

sesudo

In the [sesudo] section, the tokens determine the attributes of the sesudo utility.

echo_command

Determines whether sesudo displays the command before executing it. To echo the command, set the token value to yes.

Default: No

echo_success

Determines whether sesudo should print the successful message to the terminal when a successful sesudo command is run.

Valid values are yes and no.

Default: yes

More information:

sesudo Utility (see page 214)

standalone

In the [standalone] section, the tokens specify options for administrating using a standalone machine.

full_login_check

Specifies whether to consider administrating a site using standalone as a login.

Valid values are 0 and 1.

When this token is set to 1, it is considered as a login to the machine.

Default: 0

tcp_communication

In the [tcp_communication] section, the token defines common TCP connection settings.

listening_backlog

Defines the number of simultaneous new TCP connection requests that each listening block can establish.

Default: 128

tng

In the [tng] section, the tokens control the integration of CA Access Control into the Unicenter TNG environment.

defsesid

Specifies the default session group ID for users that do not have a specific session group ID defined.

Session groups are used by CA SSO.

Default: CAUNICENTER

ssf_numsubp

Specifies the number of subprocesses required for the sessfgate daemon to start processing incoming SSF requests.

Default: 1

sso_applname

For sites using the CA-Ticket functionality of CA SSO, specifies an eight-character string that *must* correspond to the keymgmt files found under the seos home directory in the folder data/keymgmt. The names of these files are under the SSO_APPLNAME_key.

For example, if the default value of UNICENTR is taken, the name of the file becomes UNICENTR_key.

Default: UNICENTR

The pmd.ini File

Valid on UNIX

The pmd.ini file contains various setup and initialization settings CA Access Control uses when building and maintaining a specific PMDB. It consists of several sections and each section contains multiple settings:

Section	Description
endpoint_management	Policy Model endpoint management settings.
lang	CA Access Control management interface (selang) settings for working with a Policy Model.
logmgr	PMDB logging facility settings.
passwd	User and password data settings.
pmd	Policy Model daemon (sepmdd) settings.
seos	Generic PMDB settings.

endpoint_management

The [endpoint_management] section contains the parameters that define endpoint management settings for the Policy Model.

AutoSync

Specifies to automatically synchronize the the DH with the Message Queue server.

Limits: 0,1

Default: 0 (disabled)

debug_mode

Specifies if CA Access Control writes debug messages to the endpoint_management.log file in the DMS directory (1).

Limits*:* 0,1

Default*:* 0 (debugging is disabled)

Note: The log file is located at *ACInstallDir*/log/endpoint_management.log

operation_mode

Specifies whether central (DMS) endpoint management through the CA Access Control Message Queue is enabled.

Limits: 0,1

Default: 1 (enabled)

lang

The [lang] section contains the parameters used by the CA Access Control language program (selang) when building and maintaining a PMDB.

pre_user_exit

Specifies the path of the exit program to be executed before CA Access Control issues a language command to update the UNIX user database.

post_user_exit

Specifies the path of the exit program to be executed after CA Access Control issues a language command to update the UNIX user database.

pre_group_exit

Specifies the path of the exit program to be executed before CA Access Control issues a language command to update the UNIX groups database.

post_group_exit

Specifies the path of the exit program to be executed after CA Access Control issues a language command to update the UNIX groups database.

logmgr

The [logmgr] section contains the parameters used by the PMDB logging facility.

audit_back

Specifies the name of the PMDB audit backup file.

Default: pmd_audit.bak

audit_log

Specifies the name of the PMDB audit log file.

Default: pmd_audit

audit_group

Specifies the group that can read the PMDB audit files. If no group is specified, only root can read the audit files. CA Access Control does not verify the value of this token, so if you enter an invalid group name, CA Access Control does not assign any group permissions to the audit log files.

To change the group ownership of an existing audit log file, do the following:

1. Use the selang command chgrp to set the group ownership of the files.
2. Change the UNIX permissions by entering:

```
chmod 640 /opt/CA/AccessControl//log/seos.audit
```

Default: none

audit_size

Specifies the size of the PMDB audit log file, in KB. Do not specify a size less than 50 KB.

Default: 50 KB

error_back

Specifies the name of the PMDB error backup file.

Default: pmd_error.bak

error_log

Specifies the name of the PMDB error log file.

Default: pmd_error

error_group

Specifies the group that can read the PMDB error files. If no group is specified, only root can read the error files. CA Access Control does not verify the value of this token, so if you enter an invalid group name, CA Access Control does not assign any group permissions to the error log files.

To change the group ownership of an existing error log file, do the following:

1. Use the selang command chgrp to set the group ownership of the files.
2. Change the UNIX permissions by entering:

```
chmod 640 /opt/CA/AccessControl//log/seos.error
```

Default: none

error_size

Defines the maximum size, in KB, of the PMDB error log file (defined by error_log).

Limits: A minimum value of 50 KB.

Default: 50

max_log_size

Specifies the size of the PMDB general log file in KB.

Default: 50 KB

pmd_log_level

Determines the messages that are logged in the PMDB log file.

Valid values include the following:

0-Do not log any entries.

1-List only error messages.

2-List error and informational messages.

Default: 2

use_syslog

Determines whether the policy model daemon should write syslog messages.

Default: yes

passwd

The [passwd] section contains parameters for UIDs and GIDs.

AllowedGidRange

Specifies reserved numbers.

The integers below the first number and above the second number are reserved GIDs, which CA Access Control cannot update.

Note: If only one integer is specified, all integers between one and the specified integer are reserved GIDs. If you specify a number that is larger than the upper limit, the default upper limit is applied (30000). If you specify a negative number, the default lower limit is applied (1). The applied lower limit for any number is +1 of the specified lower limit. For example, if *AllowedGidRange = 100, 3000*, then 101 is treated as the lower limit.

Limits: -1 to 2147483647

Default: 100,30000

AllowedUidRange

Specifies reserved numbers.

The integers below the first number and above the second number are reserved UIDs, which CA Access Control cannot update.

Note: If only one integer is specified, all integers between one and the specified integer are reserved UIDs. The applied lower limit for any number is +1 of the specified lower limit. For example, if *AllowedGidRange = 100, 3000*, then 101 is treated as the lower limit.

Limits: -1 to 2147483647

Default: 100,30000

pmd

The [pmd] section contains the attributes used by the sepmdd daemon when building and maintaining a PMDB.

_min_retries_

Specifies the minimum number of attempts that sepmdd should make to resend the next queued update to an unavailable subscriber. The sepmdd loops through the list of subscribers for outstanding updates and increments the counter each time it cannot resend the update to an unavailable subscriber. The subscriber is marked unavailable after the minimum number of attempts specified in this token.

Default: 4

_QD_timeout_

Specifies the maximum time, in seconds, that the sepmdd daemon waits while attempting to update a subscriber database during the first scan of its subscriber list. If the time elapses and the daemon does not succeed in updating a subscriber, it skips that particular subscriber and tries to update the remainder of the subscribers on its list.

After completing the first scan of the subscriber list, sepmdd then performs a second scan in which it attempts to update the subscribers it did not succeed in updating during the first scan. During the second scan, it tries to update a subscriber until the connect system call times out (approximately 90 seconds).

Default: 3

_retry_timeout_

Specifies the time, in minutes, to wait before trying to resend an update to an unavailable subscriber, after the minimum number of attempts specified in _min_retries_ has been made. It marks the subscriber available after the number of minutes defined by this token elapses.

A subscriber is marked unavailable until:

- It is manually released.
- sepmdd is manually shutdown and restarted. The sepmdd is restarted if:
  - if a language facility attempts to connect to it.
  - if a parent PMDB wants to send an update.
  - the pull option is triggered by a subscriber. This optionally occurs when CA Access Control starts on the subscriber.
- The pulll option is triggered by the unavailable subscriber.

Note: Shutting down sepmdd too often is not desirable because it takes time to restart the daemon, which results in slowing the whole propagation process. Allowing it to be on all the time is also undesirable because there maybe some stability issues, but it is only a conjecture.

Default: 30

_shutoff_time_

Specifies the time, in minutes of activities before sepmdd quits. If the token value is zero, sepmdd never quits.

Default: 0

always_propagate

If this token is set to no, commands that failed to execute by the policy model are not propagated to the subscribers.

Default: none

exclude_file

Specifies an exclude file.

The exclude file contains host names (one on each line) that should be excluded from receiving policy model updates.

Default: none

exclude_localhost

Tells the pmdb to exclude the local host from receiving updates as a subscriber.

Possible values: yes, no.

Default: no

exclude_method

Enables/disables the promote offset in update file when subscriber is excluded.

Values:

"pmdwait”—do not promote offset

Otherwise—"bypass"

Default: pmdwait

filter

Specifies the name of the filter file.

force_auto_truncate

Specifies whether CA Access Control truncates the update file even if there are no subscribers to the Policy Model.

You can truncate the update file manually (sepmd -t), and CA Access Control also truncates the file automatically based on a separate configuration setting (trigger_auto_truncate) that defines the event that triggers automatic truncation.

Note: If all subscribers to the Policy Model are "Out of sync", the Policy Model effectively has no subscribers.

Default: Yes

group_file_name

Specifies the name of the group file for a new UNIX group. sepmdd saves the group entry of the new UNIX group in this file.

Default: group

is_maker_checker

Specifies whether to use Dual Control. The valid values for this token are yes and no.

If **yes** is selected, then the PMDB cannot be updated directly, but only through a transaction; and each transaction entered by one administrator must be processed by another administrator before the commands are implemented on the PMDB.

Default: no

password_file_name

Specifies the name of the password file for new UNIX users. sepmdd stores the password entry of new UNIX users in this file.

Default: passwd

send_unix_env

Indicates whether sepmd sends the contents of Policy Model password files and group files.

If this token is set to **yes**, the *sepmd -n* option sends the contents of the Policy Model password files and group files.

If this token is set to **no**, the *sepmd -n* option does not send the contents of the policy model password files and group files.

Default: yes

synch_uid

Determines whether sepmdd attempts to synchronize UIDs between a Policy Model and its subscribers. The valid values for this token are yes and no.

If the token is **no**, sepmdd does not attempt to synchronize UIDs. Users are assigned the first available UID on each subscriber host.

If the token is **yes**, sepmdd attempts to synchronize UIDs. For example, if a new UNIX user is created on the PMDB with a UID of 1000, sepmdd transfers that UID to the subscribers. If UID 1000 is already in use on one of the subscribers, then the update on that subscriber fails.

sepmdd only tries to synchronize UIDs if the original command sent to the PMDB did not specify a UID for the user. If the original command did specify a UID, the specified UID is sent to all the subscribers.

Default: yes

TNG_Environment

Specifies whether the database is created with special TNG classes and resources.

Valid values are:

"0" to create the database without the special TNG classes

"1" to create the database with all the special TNG classes

Default: 0

transaction_lib

Specifies the path of the maker-checker policy.

Default: /opt/CA/eTrustAccessControl/policies/maker

trigger_auto_truncate

Defines the size of the Policy Model update file, in megabytes, that triggers an automatic truncating of the update file.

If you use a value that is less than the lower limit, CA Access Control uses the default value. If you use a value that is greater than the upper limit, CA Access Control uses the upper limit value.

Limits: 1 - 2000 MB

Default: 1024 MB

update_while_processing

Defines the frequency at which the Policy Model propagates commands to subscribers while it is processing incoming events.

The frequency is a factor of the updates_in_chunk setting, and determines how many commands the PMD processes before it sends the next subscriber in line one set of commands. For example, if you set this to 3 and updates_in_chunk is set to 10, the PMD will process 30 commands before it sends a set of commands (10) once to the next subscriber in line. A value of 0 means that the PMD does not propagate commands while processing incoming events.

Default: 1

updates_in_chunk

Determines the maximum number of commands that the Policy Model sends to each of its subscribers in each cycle of a loop.

Default: 20

UseEncryption

Specifies whether update information saved to the updates.dat file is encrypted.

Default: no

UseShadow

Determines whether to use a shadow file when you reference the PMDB native environment.

Default: no

YpServerSecure

Specifies the name of the password shadow file (a security file on an NIS server) that is used for building the NIS password map. This token is relevant only if you set UseShadow to yes.

Default: /etc/shadow

seos

The token of the [seos] section, which contains the global settings used by CA Access Control, is described in the following table.

parent_pmd

Defines a comma-separated list of policy model databases (PMDBs) from which this PMDB accepts updates. This PMDB rejects updates from any PMDB that is not specified in this list.

You can also specify a file path that contains a line-separated list of PMDBs.

Set this token to "_NO_MASTER_" for this PMDB to accept updates from any PMDB.

If you do not set this token, this PMDB does not accept updates from any PMDB.

Each PMDB is specified in the following format: pmd_name@hostname

For example:

```
parent_pmd = pmd1@host1,pmd2@host1,pmd3@host2
parent_pmd = /opt/CA/AccessControl//parent_pmdbs_file
```

Default: Token is not set (PMDB does not accept updates from any PMDB).

The lang.ini File

Valid on UNIX

This section describes the tokens in the lang.ini file, used by the selang utility.

The lang.ini file contains the following sections:

general

Contains default parameters that apply to more than one type of resource; that is, both new resources and new users.

history

Contains default parameters for the selang history mechanism.

newres

Contains the default values that are assigned to the properties of new resource records. The default value is assigned unless you explicitly set a different value.

newusr

Contains the default values that are assigned to the properties of new user records. The default value is assigned unless a different value is explicitly set.

properties

Contains tokens that specify values for user-defined properties, such as file locations for user-defined properties. The tokens have no default values; you must set them explicitly.

unix

Contains the default values that are assigned when a new user is defined to UNIX from within the selang command shell. The default value is assigned unless you explicitly set a different value.

general

The [general] section contains default parameters that apply to more than one type of resource.

defaultOwner

The name of the owner assigned to a new record.

If you do not specify a value, the creator of the new record is assigned as owner.

history

The [history] section contains default parameters for the selang history mechanism.

HistFile

The name of the file where the commands in the history list are stored. The command list is loaded at the beginning of each session.

No default value; that is, the history list is not saved at the end of a session.

HistSize

The number of commands (a positive integer between 10 and 100) stored by the history mechanism.

Default: 30

newres

The [newres] section contains default values that are assigned by the newres command. The newres command creates new resource records in the database. Each token in this section represents a newres parameter. Parameters not represented in the lang.ini file are assigned default values that are hard-coded in CA Access Control. If you do not specify a value for a token, the default value specified in the table is applied.

DefaultAudit

The default audit mode for the new resource. Valid values are: none, all, success, failure.

Default: failure

DefaultDay

The default day restrictions that apply to the resource. Valid values are: anyday, weekdays, mon, tue, wed, thu, fri, sat, sun.

Default: anyday

DefaultNotify

The default email address to which alert messages regarding the resource record are sent.

No default value; that is, no notification message is sent.

DefaultTime

The default time restrictions that apply to the resource. Valid values are: anytime, startTime:endTime.

Default: anytime

DefaultWarning

Whether warning mode is enabled by default. Valid values are: yes, no.

Default: no

newusr

The [newusr] section contains the default values assigned by the newusr command, which creates new user records in the database. Each token in this section represents a newusr parameter. Parameters not represented in the lang.ini file are assigned default values that are hard-coded in CA Access Control. If you do not specify a value for a token, the default value specified in the table is applied.

DefaultAudit

The default audit mode for the new user. Valid values are: none, all, success, failure, loginsuccess loginfailure.

Default: failure loginfailure loginsuccess

DefaultDay

The default day restrictions that apply to the user when logging in to the system. Valid values are: anyday, weekdays, mon, tue, wed, thu, fri, sat, sun.

Default: anyday

DefaultExpire

The default expiry date for the user record. Valid values are: expire[*dd*/*mm*/*yy*], expire-.

Default: expire-

DefaultLocation

The default location in which the user works.

No default value

DefaultNotify

The default email address to which alert messages are sent when the user logs in.

No default value; that is, no notification message is sent.

DefaultOrg

The organization for which the user works.

No default value

DefaultOrgUnit

The organizational unit in which the user works.

No default value

DefaultTime

The default time restrictions that apply to the user when logging in to the system. Valid values are: anytime, startTime:endTime.

Default: anytime

properties

The [properties] section contains parameters that apply to user-defined properties.

UserDefinedTokensFile

The path for a definition file that contains context information for user-defined properties.

Default: none

UserDefinedAttributesFile

The path for a definition file that contains attribute information for user-defined properties.

Default: none

User-Defined Properties

This section is complimentary to the sepropadm utility. It defines the selang context by which database properties created with sepropadm are recognized. Two definition files that use a format similar to the one used by sepropadm accomplish this. The location of these files is specified in the two tokens of this section.

Note: The properties must be defined in the database (using the sepropadm utility), before the definition files are loaded by selang. The definition files are loaded automatically when selang is run, during the initialization phase.

When these properties are defined in both the appropriate definition files and the database, you can use them in selang commands like any other CA Access Control defined property.

Important! Do **not** use the sepropadm utility with a description file that was **not** certified by your vendor's support personnel.

More information:

sepropadm Utility—Administer Database Properties (see page 199)

The Definition Files

To get selang to recognize the new user-defined properties, selang loads two *.def files during its initialization: the Tokens file and the Attributes file.

The Tokens File

User Defined Tokens File

A definition file supplied by your vendor's support personnel. The definition file has the following format:

Lines that begin with a semicolon (;) are comments and are not processed.

One line must begin with the hash symbol (#). This line must precede the description lines.

The description line must conform to the following format:

```
TOKEN=%s DOMAIN=%d CLASS=%d COMMAND=%d
```

The following is a sample definition tokens file:

```
; Sample Token Definition File for user defined properties
; Copyright 2004 Computer Associates International, Inc.
; ------------------------------------------
; DO NOT USE THIS FILE UNLESS YOU KNOW HOW TO!
# token definition file
; Format is :
TOKEN=EMAIL DOMAIN=1 CLASS=USER COMMAND=206
TOKEN=NOEMAIL DOMAIN=1 CLASS=USER COMMAND=206
TOKEN=EMAIL DOMAIN=1 CLASS=USER COMMAND=218
TOKEN=AGE DOMAIN=1 CLASS=USER COMMAND=206
TOKEN=AGE DOMAIN=1 CLASS=USER COMMAND=218
TOKEN=NOAGE DOMAIN=1 CLASS=USER COMMAND=206
TOKEN=TERMLOCATION DOMAIN=1 CLASS=TERMINAL COMMAND=217
TOKEN=NOTERMLOCATION DOMAIN=1 CLASS=TERMINAL COMMAND=205
TOKEN=TERMLOCATION DOMAIN=1 CLASS=TERMINAL COMMAND=205
```

The Attributes File

User Defined Attributes File

A definition file supplied by your vendor's support personnel. The definition file has the following format:

Lines that begin with a semicolon (;) are comments and are not processed.

One line must begin with the hash symbol (#). This line must precede the description lines.

The description line must conform to the following format:

```
PROPERTY=%s TYPE=%d FLAGS=%x
```

The following is a sample definition attributes file:

```
; Sample Attributes Definition File for user defined properties
; Copyright 2004 Computer Associates International, Inc.
; ----------------------------------------
; DO NOT USE THIS FILE UNLESS YOU KNOW HOW TO!
# attributes definition file
; Format is :
PROPERTY=EMAIL TYPE=306 FLAGS=8000
PROPERTY=EMAIL TYPE=5 FLAGS=8000
PROPERTY=AGE TYPE=306 FLAGS=8000
PROPERTY=AGE TYPE=5 FLAGS=8000
PROPERTY=TERMLOCATION TYPE=306 FLAGS=8000
PROPERTY=TERMLOCATION TYPE=5 FLAGS=8000
```

Important! Do **not** use selang with a definition file that was **not** certified by your vendor's support personnel.

unix

The [unix] section contains the default values that are assigned by the newusr command when a user is added to UNIX. Each token in this section represents an argument of the *unix* parameter. UNIX arguments not represented in the lang.ini file are assigned default values that are hard-coded in CA Access Control.

DefaultPGroup

The default group assigned to new users. If you specify a default shell in the server's seos.ini file, it overrides the value specified here.

Default: other

DefaultShell

The default shell of new users. If you specify a default shell in the server's seos.ini file, it overrides the value specified here.

Default: /bin/sh

DefaultHome

The default home directory of the system. If you specify a default shell in the server's seos.ini file, it overrides the value specified here. The user's home directory is a subdirectory of the specified system home directory. For example, if the system home directory is /home, the new user's home directory is /home/userName. If you specify a home directory prefix in the server's seos.ini file, it overrides the value specified here.

For those familiar with earlier versions, the token DefaultHome replaces HomeDirPrefix.

Default: /home

trcfilter.init

Valid on UNIX

The CA Access Control daemon also uses the trcfilter.init initialization file.

This optional file contains entries that specify filter masks for filtering out CA Access Control trace messages. Each line of the file contains a regular expression. When a message is sent to the trace file, seosd checks whether the message matches one of the entries in the trcfilter.init file. It writes the trace message to the file only if it does not match any of the expressions specified in the trcfilter.init file.

For example, the following trcfilter.init file causes all messages that begin with “INFO” or “WATCHDOG” to be discarded. They are not written to the trace file.

```
WATCHDOG*
INFO*
```

Note: This file does not filter audit records generated by user traces. To filter these audit records, edit the audit.cfg file.

audit.cfg File—Filter Audit Records

The audit.cfg file filters audit records on a host by defining records that are not sent to the audit file. Each line represents a rule for filtering out audit information.

By default, the audit.cfg file is located in the following directories:

- (UNIX) /opt/CA/AccessControl/etc
- (Windows) C:\\ProgramFiles\CA\AccessControl\data

You can change the location of the audit.cfg file by editing the [logmgr] AuditFiltersFile token in the seos.ini file (UNIX), or the AuditFiltersFile entry in the logmgr registry key (Windows).

Note: Save the audit.cfg file using UTF-8 encoding if you filter the file that includes Japanese characters.

Use the audit.cfg file to filter out records in the following audit event types, each type by a different syntax:

- resource access
- network connection (see page 391)
- login and logout events (see page 392)
- security database administration (see page 394)
- trace message on a user

Note: A * in any column in each type of syntax stands for "any value".

audit.cfg File—Resource Access Events Filter Syntax

Audit records that belong to a resource access event have the following filter format:

ClassName;*ObjectName*;*UserName*;*ProgramPath*;*Access*;*AuthorizationResult*

ClassName

Defines the name of the class that the accessed object belongs to.

Note: Enter the name of the class in uppercase.

ObjectName

Defines the name of the object that was accessed.

UserName

Defines the name of the accessor.

ProgramPath

Defines the name of the program used to access the object.

Access

Defines the requested access to the object.

Note: The following values are the values for this parameter that you use in the audit.cfg file to filter out an audit record. In some cases the value of this parameter in the audit.cfg file is different to the value that CA Access Control writes in the audit record for that event. Any such differences are noted after the description of each value. Type the parameter in the same case as it appears in the following list.

Values:

*****

A wildcard that represents any type of access.

Chdir

Change directory—The accessor made a request to move the object to a different directory.

Chmod

Change mode—The accessor made a request to change the mode of the object.

Chgrp

(UNIX) Change group—The accessor made a request to change the group the object belongs to.

Chown

Change owner—The accessor made a request to change the owner of the object.

Connect

Join user to group—The accessor made a request to add a new user to a group.

Note: The connect value is identical to the join value.

Control

(UNIX) Control—The accessor requested Chown, Chmod, Utime, Sec, Chdir, and Update access to the object.

Cre

Create—The accessor made a request to create an object.

Crrdwr

Create, Read, and Write—The accessor requested Create, Read, and Write access to the object.

Note: CA Access Control writes this value as CrRdWrite in the corresponding audit record.

Crread

Create and Read—The accessor requested Create and Read access to the object.

Note: CA Access Control writes this value as CrRead in the corresponding audit record.

Crwrite

Create and Write—The accessor requested Create and Write access to the object.

Note: CA Access Control writes this value as CrWrite in the corresponding audit record.

Del

Delete—The accessor made a request to delete an object.

Note: CA Access Control writes this value as Erase in the corresponding audit record.

Filereplace

Create and Erase—The accessor requested Create and Erase access to the object.

Note: CA Access Control writes this value as Replace in the corresponding audit record.

Filescan

Filescan—The accessor requested List access to the object.

Note: CA Access Control writes this value as Scan in the corresponding audit record.

Join

Join user to group—The accessor made a request to add a new user to a group.

Note: The join value is identical to the connect value.

Kill

Kill—The accessor made a request to kill a process.

Modify

Modify—The accessor requested Modify access to the object.

OwnGrp

Change owner and Change group—The accessor requested Chown and Chgrp access to the object.

PW

Password—The accessor made a request to change a password.

Note: CA Access Control writes this value as Password in the corresponding audit record.

R

Read—The accessor requested read access to an object.

Note: (UNIX) If STAT_intercept is set to 1, this parameter includes *stat* interception.

Rename

Change file name—The accessor made a request to change the file name of an object.

Sec

Change ACL—The accessor made a request to edit the ACL of the object.

Note: CA Access Control writes this value as ACL in the corresponding audit record.

Update

Read, Write, and Execute—The accessor requested Read, Write, and Execute access to an object.

Note: The Update value also filters events when an accessor requested Read and Write access to an object.

Utime

(UNIX) Change time—The accessor made a request to change the modification time of an object.

Note: CA Access Control writes this value as Utimes in the corresponding audit record.

W

Write—The accessor requested write access to an object.

X

Execute—The accessor made a request to execute an object.

Note: Some values are not valid for every class. For example, kill is an invalid value for the FILE class, because the kill action is not available to objects in the FILE class. If you enter an invalid value for a class when you write a rule, CA Access Control ignores that rule when it reads the file.

AuthorizationResult

Defines the authorization result.

Values: P (permitted), D (denied), *

Example: Audit Filter Policy

- This example shows you what an audit filtering policy looks like:

```
env config
er config audit.cfg line+("FIEL;*;*;*;R;P")
```

- This policy writes the following line to the audit.cfg file. The line filters audit records that record a permitted attempt by any accessor to access any file resource for reading:

```
FILE;*;*;*;R;P
```

audit.cfg File—Network Connection Events Filter Syntax

Audit records that belong to a network connection event have the following filter format:

{HOST|TCP};*ObjectName*;*HostName*;*ProgramPath*;*Access*;*AuthorizationResult*

HOST

Specifies that the rule filters records generated by objects in HOST class, that is, incoming TCP connections.

TCP

Specifies that the rule filters records generated by objects in TCP class, that is, connect with service events.

ObjectName

Defines the name of the object that was accessed. *ObjectName* can be a service name or port number.

HostName

Defines the name of the host. *HostName* must be an object in the HOST class.

ProgramPath

Defines the login program type.

(Windows) For outgoing connections, this parameter defines the program path of the process trying to establish the connection.

Note: This parameter has no meaning for incoming connection events. Use * for this parameter to filter audit records generated by incoming connection events.

Access

Defines the type of attempted connection.

Values:

- (HOST) *
- (TCP) R (incoming connection), W (outgoing connection), *

AuthorizationResult

Defines the authorization result.

Values: P (permitted), D (denied), *

Examples: Filter Network Connection Events

- This example filters all audit records from the host ca.com generated by successful incoming telnet connections:

  ```
  HOST;telnet;ca.com;*;*;P
  ```

- This example filters all audit records from the host ca.com generated by incoming and outgoing login TCP connections that were denied:

  ```
  TCP;login;ca.com;*;*;D
  ```

- This example filters all audit records from the host ca.com generated by outgoing telnet connections:

  ```
  TCP;telnet;ca.com;*;W;*
  ```

audit.cfg File—Login and Logout Events Filter Syntax

Audit records that belong to a login or logout event have the following filter format:

LOGIN;*UserName*;*UserId*;*TerminalName*;*LoginProgram*;*AuthorizationResultOrLoginType*

LOGIN

Specifies that the rule filters audit records generated by login and logout events.

UserName

Defines the name of the accessor.

UserId

(UNIX) Defines the native user ID of the accessor.

TerminalName

Defines the terminal at which the event occurred.

LoginProgram

Defines the name of the program that attempted to log in or out.

AuthorizationResultorLoginType

Defines the authorization result.

Values:

*****

A wildcard that represents any type of authorization result.

D

The login attempt was denied.

P

The login attempt was permitted.

O

(UNIX) The accessor logged out.

I

(UNIX) The serevu daemon revoked the accessor's account.

E

(UNIX) The serevu daemon enabled the accessor's account.

A

(UNIX) The serevu daemon or Pluggable Authentication Module audited a user's attempt to log in with an incorrect password.

Note: Windows does not record logout events.

Examples: Filter Login or Logout Events

- This example filters all audit records generated when root logs in to a permitted account:

  ```
  LOGIN;root;*;*;*;P
  ```

- This example filters all audit records generated when root logs in successfully due to the system's CRON program:

  ```
  LOGIN;root;*;*;SBIN_CRON;P
  ```

- This example filters all audit records generated when the _CRONJOB_ process logs the root user out:

  ```
  LOGIN;root;*;_CRONJOB_;*;O
  ```

audit.cfg File—Security Database Administration Events Filter Syntax

Audit records that belong to a security database administration event have the following filter format:

```
ADMIN;ClassName;ObjectName;UserName;EffectiveUserName;TerminalName;Command;CommandResult
```

ADMIN

Specifies that the rule filters audit records generated by events performed by an administrator.

ClassName

Defines the class on which the administrator executes the command.

ObjectName

Defines the object that the administrator's command updated.

UserName

Defines the name of the user who executed the command.

EffectiveUserName

(UNIX) Defines the name of the effective user to which the rule applies.

(Windows) Defines the name of the native user to which the rule applies.

TerminalName

Defines the terminal at which the event occurred.

Command

Defines the selang command that the administrator executed.

CommandResult

Defines the authorization or command result.

Values: S (command succeeded), F (command failed), D (command denied), *

Example: Filter Security Database Administration Events

This example filters all audit records generated by successful FILE management commands by admin01:

```
ADMIN;FILE'*;admin01;*;*;*;S
```

audit.cfg File—Trace Messages On a User Events Filter Syntax

Audit records that belong to a trace message on a user event have the following filter format:

TRACE;*TracedClassName;TracedObjectName;RealUserName;EffectiveUserName;ACUserName;AuthorizationResult;TraceMessage*

Note: The maximum limit for the trace filter is 1000 records.

TRACE

Specifies that the rule filters user trace records.

TracedClassName

Defines the name of the object class the user tried to access.

Note: Enter the name of the class in uppercase.

TracedObjectName

Defines the name of the object that the user tried to access.

RealUserName

(UNIX) Defines the name of the real user that generated the trace record.

(Windows) Defines the name of the native user that generated the trace record.

EffectiveUserName

(UNIX) Defines the name of the effective user that generated the trace record.

(Windows) Defines the name of the native user that generated the trace record. This parameter is identical to the RealUserName parameter. Use * for this parameter.

ACUserName

Defines the user name CA Access Control chose to authorize the event.

AuthorizationResult

Defines the authorization result.

Values: P (permitted), D (denied), *

TraceMessage

Defines the trace message that was generated.

Example: Filter Trace On a User Message Events

This example filters all user trace records generated when the effective user is root, and root accessed an object in the FILE class:

```
TRACE;FILE;*;*;root;*;*;*
```

auditrouteflt.cfg File—Filter Audit Records Routing

The auditrouteflt.cfg file filters audit records routing by defining records that CA Access Control should not send to the Distribution Server. Each line represents a rule for filtering out audit information. The file pathname is defined by the audit_filter configuration setting in the ReportAgent section.

Note: Filtered audit events are written to the local audit file but CA Access Control does not send them to the message queue on the Distribution Server. To filter out audit messages from the local audit file, modify filter rules in the file defined by the AuditFiltersFile configuration setting in the logmgr section (by default, audit.cfg).

You can use the auditrouteflt.cfg file to filter out records in the following audit event types, each type by a different syntax:

- resource access
- network connection
- login and logout events
- security database administration
- trace message on a user

Note: A * in any column in each type of syntax stands for "any value".

Resource Access Events Filter Syntax

Audit records that belong to a resource access event have the following filter format:

ClassName;*ObjectName*;*UserName*;*ProgramPath*;*Access*;*AuthorizationResult*

ClassName

Defines the name of the class that the accessed object belongs to.

Note: You must enter the name of the class in uppercase.

ObjectName

Defines the name of the object that was accessed.

UserName

Defines the name of the accessor.

ProgramPath

Defines the name of the program used to access the object.

Access

Defines the requested access to the object.

Values:

*****

A wildcard that represents any type of access.

Chdir

Change directory—The accessor made a request to move the object to a different directory.

Chmod

Change mode—The accessor made a request to change the object's mode.

Chgrp

(UNIX) Change group—The accessor made a request to change the group the object belongs to.

Chown

Change owner—The accessor made a request to change the owner of the object.

Cre

Create—The accessor made a request to create a new object.

Del

Delete—The accessor made a request to delete an object.

Join

Join user to group—The accessor made a request to add a new user to a group.

Kill

Kill—The accessor made a request to kill a process.

R

Read—The accessor requested read access to an object.

Note: (UNIX) This parameter includes *stat* interception if STAT_intercept is set to 1.

Rename

Change file name—The accessor made a request to change the file name of an object.

Sec

Change ACL—The accessor made a request to edit an object's ACL.

Utime

(UNIX) Change time—The accessor made a request to change the modification time of an object.

W

Write—The accessor requested write access to an object.

X

Execute—The accessor made a request to execute an object.

Note: Some values are not valid for every class. For example, kill is an invalid value for the FILE class, because the kill action is not available to objects in the FILE class. If you enter an invalid value for a class when you write a rule, CA Access Control ignores that rule when it reads the file.

AuthorizationResult

Defines the authorization result.

Values: P (permitted), D (denied), *

Network Connection Events Filter Syntax

Audit records that belong to a network connection event have the following filter format:

{HOST|TCP};*ObjectName*;*HostName*;*ProgramPath*;*Access*;*AuthorizationResult*

HOST

Specifies that the rule filters records generated by objects in HOST class, that is, incoming TCP connections.

TCP

Specifies that the rule filters records generated by objects in TCP class, that is, connect with service events.

ObjectName

Defines the name of the object that was accessed. *ObjectName* can be a service name or port number.

HostName

Defines the name of the host. *HostName* must be an object in the HOST class.

ProgramPath

Defines the login program type.

(Windows) For outgoing connections, this parameter defines the program path of the process trying to establish the connection.

Note: This parameter has no meaning for incoming connection events. Use * for this parameter to filter audit records generated by incoming connection events.

Access

Defines the type of attempted connection.

Values:

- (HOST) *
- (TCP) R (incoming connection), W (outgoing connection), *

AuthorizationResult

Defines the authorization result.

Values: P (permitted), D (denied), *

Login and Logout Events Filter Syntax

Audit records that belong to a login or logout event have the following filter format:

`LOGIN;UserName;UserId;TerminalName;LoginProgram;AuthorizationResultOrLoginType`

LOGIN

Specifies that the rule filters audit records generated by login and logout events.

UserName

Defines the name of the accessor.

UserId

Defines the native user ID of the accessor.

TerminalName

Defines the terminal at which the event occurred.

LoginProgram

Defines the name of the program that attempted to log in or out.

AuthorizationResultorLoginType

Defines the authorization result.

Values:

*****

A wildcard that represents any type of authorization result.

D

The login attempt was denied.

P

The login attempt was permitted.

O

(UNIX) The accessor logged out.

I

(UNIX) The serevu daemon revoked the accessor's account.

E

(UNIX) The serevu daemon enabled the accessor's account.

A

(UNIX) The serevu daemon or Pluggable Authentication Module audited a user's attempt to log in with an incorrect password.

Note: Windows does not record logout events.

Security Database Administration Events Filter Syntax

Audit records that belong to a security database administration event have the following filter format:

ADMIN;*ClassName*;*ObjectName*;*UserName*;*EffectiveUserName*;*TerminalName*;*Command*;*CommandResult*

ADMIN

Specifies that the rule filters audit records generated by events performed by an administrator.

ClassName

Defines the class on which the administrator executes the command.

ObjectName

Defines the object that the administrator's command updated.

UserName

Defines the name of the user who executed the command.

EffectiveUserName

(UNIX) Defines the name of the effective user to which the rule applies.

(Windows) Defines the name of the native user to which the rule applies.

TerminalName

Defines the terminal at which the event occurred.

Command

Defines the selang command that the administrator executed.

CommandResult

Defines the authorization or command result.

Values: S (command succeeded), F (command failed), D (command denied), *

Trace Messages On a User Events Filter Syntax

Audit records that belong to a trace message on a user event have the following filter format:

TRACE;*TracedClassName;TracedObjectName;RealUserName;EffectiveUserName;ACUserName;AuthorizationResult;TraceMessage*

TRACE

Specifies that the rule filters user trace records.

TracedClassName

Defines the name of the object class the user tried to access.

Note: You must enter the name of the class in uppercase.

TracedObjectName

Defines the name of the object that the user tried to access.

RealUserName

(UNIX) Defines the name of the real user that generated the trace record.

(Windows) Defines the name of the native user that generated the trace record.

EffectiveUserName

(UNIX) Defines the name of the effective user that generated the trace record.

(Windows) Defines the name of the native user that generated the trace record. This parameter is identical to the RealUserName parameter. Use * for this parameter.

ACUserName

Defines the user name CA Access Control chose to authorize the event.

AuthorizationResult

Defines the authorization result.

Values: P (permitted), D (denied), *

TraceMessage

Defines the trace message that was generated.

Examples: Filter Network Connection Events

- This example filters all audit records from the host ca.com generated by successful incoming telnet connections:

  ```
  HOST;telnet;ca.com;*;*;P
  ```

- This example filters all audit records from the host ca.com generated by incoming and outgoing login TCP connections that were denied:

  ```
  TCP;login;ca.com;*;*;D
  ```

- This example filters all audit records from the host ca.com generated by outgoing telnet connections:

  ```
  TCP;telnet;ca.com;*;W;*
  ```

Examples: Filter Login or Logout Events

- This example filters all audit records generated when root logs in to a permitted account:

  ```
  LOGIN;root;*;*;*;P
  ```

- This example filters all audit records generated when root logs in successfully due to the system's CRON program:

  ```
  LOGIN;root;*;*;SBIN_CRON;P
  ```

- This example filters all audit records generated when the _CRONJOB_ process logs the root user out:

  ```
  LOGIN;root;*;_CRONJOB_;*;O
  ```

Example: Filter Security Database Administration Events

This example filters all audit records generated by successful FILE management commands by admin01:

```
ADMIN;FILE'*;admin01;*;*;*;S
```

Example: Filter Trace On a User Message Events

This example filters all user trace records generated when the effective user is root, and root accessed an object in the FILE class:

```
TRACE;FILE;*;*;root;*;*;*
```

Example: Audit Filter Policy

This example shows you what an audit filtering policy looks like:

```
env config
er config auditrouteflt.cfg line+("FILE;*;*;R;P")
```

This policy writes the following line to the auditrouteflt.cfg file:

```
FILE;*;*;R;P
```

This line filters audit records that record a permitted attempt by any accessor to access any file resource for reading.

The Audit Log Route Configuration File selogrd.cfg

Valid on UNIX

The following is the format of the configuration file, followed by a detailed explanation.

```
section-name-1
routing-method destination
[{include|exclude} match-field(match-pattern) ...]
...
.
section-name-2
routing-method destination
[{include|exclude} match-field(match-pattern) ...]
...
.
...
```

Specifying Audit Records

The configuration file is a list of which audit records to route-and which not to route-to various destinations. To specify audit records, you describe the contents of one or more particular fields. You can use the standard UNIX pattern matching (the wildcards * and ?).

For example, to specify records that deal with users whose user names begin with the letters dbms, you would enter the following:

```
User(dbms*)
```

This example matches users with names like dbms1, dbms_mgr, and so on.

To specify the same users, but only the records that deal with their login attempts, you would enter:

```
User(dbms*) Class(LOGIN)
```

Note: When a line specifies records in terms of more than one field, it specifies only the records that match *all* those fields.

At the beginning of the same line that specifies the records, you specify whether you want the records included or excluded. For example, to include those records in the routing enter the following:

```
include User(dbms*) Class(LOGIN).
```

This type of line appears in the overall format as:

```
[{include|exclude} match-field(match-pattern) ... .]
```

Here, the "..." means that the first match-field(match-pattern) pair can be followed by further pairs.

You can use any of the following for match-field(match-pattern):

Access(*access-type*)

For the type of access required; *access-type* is any one of the following:

ACL, Chdir, Chgrp, Chmod, Chown, Connect, Control, Create, Erase, Exec, Kill, Modify, Owngrp, Password, Read, Rename, Replace, Update, Utimes, and Write.

Class(LOGIN)

For login records.

Class(LOGOUT)

For logout records.

Class(PWCHANGE)

For password administration.

Class(HOST)

For TCP/IP records.

Class(UPDATE *CA Access Control-class*)

For database administration. CA Access Control-class is any of the accessor or resource classes (such as USER, GROUP, FILE, HOSTNP...) or a pattern for the class name to match. Thus for all database administration, you can specify UPDATE *.

Class(*CA Access Control-class*)

For access to protected resources. For example, Class(FILE) refers to records reporting file access attempts.

Note that you can use an asterisk to combine Class(*CA Access Control-class*) and Class(UPDATE *CA Access Control-class*) as *Class(*CA Access Control-class*)*. For example, specifying Class(*FILE) is like specifying both Class(FILE) and Class(UPDATE FILE). It refers both to attempts to access files and to attempts to update records in the FILE class.

Code(*return-code*)

For the CA Access Control return code indicating what happened; return-code can take the following values. (See also Example 1 in this section.)

A-An attempt to log in failed because an invalid password was entered repeatedly.

D-CA Access Control denied access to a resource, did not permit a login, or did not permit an update to the database because the accessor did not have sufficient authorization.

E-Serevu enabled a disabled user account.

F-An attempt to update the database failed.

I-Serevu disabled a user account.

M-The executed command started or shut down a daemon.

O-A user logged out.

P-CA Access Control permitted access to a resource or permitted a login.

S-The database was successfully updated.

T-An audit record was written because all the actions of the user are being traced.

U-A Trusted program (setuid or setgid) was changed; therefore it is no longer Trusted.

W-Access to the resource violated the access rules for the resource. However, CA Access Control allowed the access because warning mode is set in the resource.

Host(*host-name*)

For the host involved in a TCP/IP connection.

Object(*resource-name*)

For the resource that the user is attempting to access.

Reason(*reason-number*)

For the reason that the audit record is triggered.

Service(*service-name*)

For the name of the service requested from the remote host, such as telnet or ftp.

Source Host(*hostname*)

For the name of the host that contributed the record to the consolidated audit.

Stage(*stage-number*)

For the stage at which access was granted or denied. (See the lists of stage codes in the *Reference Guide*.)

Terminal(*terminal-name*)

For the terminal that is attempting access or administration.

Uid(*uid-number*)

For the uid of the user who is attempting access or administration.

User(*username*)

For users attempting access or administration; username is a name or pattern.

Note: Although some variables are more likely to be specified as patterns, you can use a pattern for any variable-even for something like a stage number.

Refining with Further Lines

To refine your specifications, you can filter by differing criteria at the same time. Simply add one include/exclude line after another. For example:

```
include User(dbms*) Class(*LOGIN*).
exclude Terminal(console_*).
```

The example specifies all login attempts by users whose names begin with dbms and who are at terminals that do not have names beginning with console_.

Specifying the Destination

Use a line *above* your sequence of include and exclude lines to specify the destination for the audit records you are including. For example:

```
mail weekwatch
include User(dbms*) Class(*LOGIN*).
exclude Terminal(console_*).
```

The example specifies that the email address weekwatch receives a report on all login attempts by users whose names begin with dbms and who are at terminals that do not have names beginning with console_.

This type of line appears in the format of the log route configuration file as:
routing-method destination

You can use any of the following methods:

mail *address*

To email the audit record; *address* is the destination address. If it is not in the form user@host, it is checked against local user lists and the NIS mail alias map.

Note: If address is a user name and surrogate requests to that user's account are audited, the audit records accumulate endlessly.

screen *username*

To display the audit record on the screen of the specified user, if that user is logged in at the current host when selogrd forwards the audit record. If the user is not logged in, the display is canceled, not postponed.

cons *hostname*

To send the audit record to the Security Administrator GUI of the secmon utility on the specified host. If that host is not available, the display is terminated, not postponed.

file *textfilename*

To write the audit record in the specified ASCII file; *textfilename* must be an absolute path name and selogrd must have access to the file.

host *hostname*

To send the audit record to the audit log collector on the specified host. If that host is not available, selogrd tries again later.

notify mail *or* notify default

To email the audit record to the address that the audit record itself specifies.

notify screen

To display the audit record on the screen of the user that the audit record itself specifies. If the user is not logged on, the display is canceled, not postponed.

syslog *priority*

To send the audit records to the syslog with a specified log priority:

- **LOG_EMERG**—System is unusable.
- **LOG_ALERT**—Action must be taken immediately.
- **LOG_CRIT**—Critical conditions.
- **LOG_ERR**—Error conditions.
- **LOG_WARNING**—Warning conditions.
- **LOG_NOTICE**—Normal but significant condition.
- **LOG_INFO**—Informational.
- **LOG_DEBUG**—Debug-level messages.

uni hostname

To send the audit record to the Unicenter TNG event manager on the specified host. You must also set selogrd to load the uni.so shared library, which is found in the *ACInstallDir*/lib directory. Note that the installation performs this task for you if it finds Unicenter TNG installed on the specified host and you choose to do it.

Proper Sequence for Lines

It is important to arrange your include and exclude lines in proper sequence, properly delimited.

- You must precede each sequence of lines (or single line) that you want to treat as a single complex filter with a title line, and end it with a terminating line that consists of a single dot; for example:

  dbms login from non-console

  ```
  mail weekwatch
  ```

  include User(dbms*) Class(*LOGIN*).

  ```
  exclude Terminal(console_*).
  .
  ```

The full sequence, including the title line and terminating line, is called a *section* of the file.

- If both include and exclude lines match the same audit record in the same section, the last match overrides all others.
- If no lines match a particular audit record, then the first line of the section is the deciding line for that record. (If the first line is an include line, then the failure to match excludes the record. If the first line is an exclude line, then the failure to match includes the record for routing.)
- If the section includes no include and exclude lines, then it includes all audit records for routing.

How Sections Coexist

Whereas the lines of a section work together to produce a single decision as to whether or not a record is to be sent, different sections in the configuration file work entirely independently. Whether or not an audit record is sent by one section, has no influence on whether the same audit record is sent by another section.

You can send the same selection of audit records to more than one destination, and the same destination can receive more than one selection of audit records.

In your configuration file, the total of all the include and exclude lines-from all the sections together-must not exceed 64 lines.

Including Comments

To add a comment line to the configuration file, begin the line with a semicolon.

Example 1

The following is a sample configuration file, followed by its explanation.

```
; Product : CA Access Control
; Module  : selogrd
; Purpose : route table for audit log routing daemon
;
;-----------------------------------------------------
 Rule#1
 mail jones@admhost
 include       Class(*LOGIN*) Code(D).
 .
 Rule#2
 mail smith
 include       Class(*SURROGATE*) Object(USER.root*).
 .
 Rule#3
 host venus
 exclude       Class(UPDATE SU*).
 .
 Rule#4
 host venus
 include       Class(*PROGRAM*) Object(/usr/bin/ps).
 .
```

The first five lines are comment lines.

The next four lines make up the first section, named Rule#1. They tell selogrd to mail a log record to the address jones@admhost whenever a login request is denied (code D reports denial):

```
Rule#1
mail jones@admhost
include       Class(*LOGIN*) Code(D).
.
```

The next section is named Rule#2. It tells selogrd to mail a log record to the address smith whenever someone attempts to use the su command to enter the root account (the objects in the SURROGATE class are targets for the su command):

```
Rule#2
mail smith
include       Class(*SURROGATE*) Object(USER.root*).
.
```

The next section is named Rule#3. It tells selogrd to send a log record to the collector on host venus whenever someone attempts database administration, unless the class name begins with the letters SU (the matching classes are SURROGATE and SUDO):

```
Rule#3
host venus
exclude       Class(UPDATE SU*).
```

The last section is named Rule#4. It tells selogrd to send a log record to the collector on host venus whenever someone attempts to use the ps command:

```
(Code 1 8pt) Rule#4
host venus
include       Class(*PROGRAM*) Object(/usr/bin/ps).
.
```

Example 2

The following configuration file sends *all* audit records to the collector on the station named loghost:

```
 ; Product : CA Access Control
 ; Module  : selogrd
 ; Purpose : route table for audit log routing daemon
 ;
 ;-------------------------------------------------------
  Rule#1
  host loghost
  .
```

Return Codes

You can associate each type of record in the configuration file with one or more CA Access Control return code. (For a complete list of the return codes see the description of *code(return-code*) in Specifying Audit Records in this section.) The following table describes the record types and their associated return codes.

Record Type	Class or Event	Associated Return Codes
Login	LOGIN	D, P, W
	LOGINDISABLE	I
	LOGINENABLE	E
Logout	LOGOUT	O
TCP/IP	HOST	D, P
Resource classes	*Class name*	D, P, W
Watchdog	PROGRAM	U
	SECFILE	U
Password administration	PWCHANGE	D
Down	SHUTDOWN	D, S
Start	START	S
CA Access Control database administration	UPDATE	D, F, S

The uxauth.ini File

Valid on UNIX

The uxauth.ini configuration file contains various tokens that control the functionality of UNIX Authentication Broker. The UNIX Authentication Broker configuration file is divided into sections that relate to different sets of tokens that control UNIX Authentication Broker functionality:

Section	Description
ad	Contains Active Directory tokens with the parameters that you entered during installation
agent	Contains tokens that control the various UNIX Authentication Broker parameters
global	Contains tokens that control UNIX Authentication Broker general settings

Section	Description
libdefaults	Contains tokens that control Kerberos configuration settings
logmgr	Contains tokens that the UNIX Authentication Broker logging utility uses
map	Contains tokens that specify Active Directory attribute names
message	Contains tokens that UNIX Authentication Broker uses to define the message file
migrate	Contains tokens that UNIX Authentication Broker uses during the migration process
pam	Contains tokens that control the UNIX Authentication Broker PAM module
passwd	Contains tokens that UNIX Authentication Broker uses to control password changes during the migration process
register	Contains tokens that control the UNIX Authentication Broker registration functionality

ad

The [ad] section contains Active Directory tokens with the parameters that you entered during installation.

ad_domain

Defines the name of the Active Directory domain.

Note: Do not manually edit the value of this configuration setting. Use the uxconsole -register utility to set the value of this configuration setting.

ad_gc_port

Specifies the port that the Active Directory Global Catalog service uses.

Default: 3268

ad_site

Defines the name of the Active Directory site that contains the DCs that the UNIX host uses to communicate with Active Directory.

Any values in the lookup_dc_list override the value of this configuration setting. The UNIX host does not communicate with any DC listed in the ignore_dc_list configuration setting.

Note: Do not manually edit the value of this configuration setting. Use the uxconsole -register utility to set the value of this configuration setting.

Default: none

base_dn

Defines the base_dn of the Active Directory server. CA Access Control automatically sets the value of this configuration setting.

cache_cleanup_interval

Specifies the cleanup interval, in hours to clean up the local users and group cache for users that are removed from partner domains with one-way trust with the registered domain. This parameter is ignored if the registration domain has no partners with one-way trust.

Value: Any positive integer.

Default: 24

Example: cache_cleanup_interval = 24

cache_cleanup_startup_time

Specifies the start time to clean up the local users and group cache for users that are removed from partner domains with one-way trust with the registered domain. This parameter is ignored if the registration domain has no partners with one-way trust.

Value: Any integer from 0 through 23.

Default: 3 (cleanup starts at 3am)

Example: cache_cleanup_startup_time = 3

computer_container

Defines the location of the UNIX host in Active Directory.

Default: cn=Computers

domain_query_order

Specifies the order in which UNIX Authentication Broker queries Active Directory domains for users and groups.

Options: none-no order specified; comma separated list of Active Directory domains

Default: none

group_container

Specifies the base entry to search for UNIX users in Active Directory.

Limits: container name (cn=groups), ROOT for the complete Active Directory query.

Default: ROOT

group_custom_filter

Specifies a custom search filter to apply during groups search in Active Directory.

Example: gidNumber=*

Default: none

ignore_dc_list

Specifies the Active Directory domain controllers that are ignored for LDAP connection.

Options: none, comma separated list of fully qualified host names

Default: none

ignore_domain_list

Specifies the Active Directory domains that UNIX Authentication Broker ignores when it queries users and groups.

Options: none - query current and all trusted domains; all - do not query trusted domains; a comma separated list of domains to ignore.

Default: none

ignore_group_container

Specifies the Active Directory group containers to ignore. Containers are defined by their Distinguished Names, comma separated.

Limits: none, comma separated list of distinguished names

Default: none

ignore_user_container

Specifies the Active Directory user containers to ignore. Containers are defined by their Distinguished Names, comma separated.

Limits: none, comma separated list of distinguished names

Default: none

ldap_port

Defines the port the Active Directory LDAP service uses.

Default: 389

lookup_dc_list

Specifies the Active Directory domain controllers that are used for LDAP connection. If you specify a list of domain controllers, UNIX Authentication Broker uses the specified domain controllers only. If you do not specify the DCs to use, UNIX Authentication Broker discovers the Active Directory site that is closest to the physical location of the endpoint and communicates with DCs in the discovered site.

Options: none, comma separated list of fully qualified host names.

Default: none

lookup_domain_list

Specifies the Active Directory domains that established a bi-directional trust with the domain that you registered UNIX Authentication Broker.

Options: none,UNIX Authentication Broker automatically discovers the trusted domains, comma separated list of trusted domains

Default: none

user_container

Specifies the base entry to search for UNIX users in Active Directory.

Limits: container name, ROOT for complete Active Directory query.

Default: ROOT

user_custom_filter

Specifies a custom search filter to apply during users search in Active Directory.

Default: none

agent

The [agent] section contains tokens that control the various UNIX Authentication Broker parameters.

ac_registration_interval

Defines the interval, in seconds, for registering UNIX Authentication Broker with the CA Access Control endpoint. A value of 0 specifies no registration.

Default: 60

Note: UNIX Authentication Broker attempts to register with the endpoint only if CA Access Control is installed on the UNIX host.

ad_group_deny_gid_list

Defines the GIDs (comma-separated) of Active Directory groups that cannot log in.

Example: ad_group_deny_gid_list = 11,14

Note: This parameter is valid in full integration mode only.

Default: Token not set (no default)

ad_group_minimal_gid

Defines the minimal GID of Active Directory groups that can log in.

Note: This parameter is valid in full integration mode only.

Default: Token not set (no default)

ad_user_deny_uid_list

Defines the UIDs (comma-separated) of Active Directory users that cannot log in.

Example: ad_user_deny_uid_list = 12,37

Note: This parameter is valid in full integration mode only.

Default: Token not set (no default)

ad_user_minimal_uid

Defines the minimal UID of Active Directory users that can log in.

Note: This parameter is valid in full integration mode only.

Default: Token not set (no default)

agent_open_files_max

Specifies the maximum number of opened files, that the UNAB agent can use. The UNAB agent restarts if it exceeds the maximum value.

Default: 100

Example: agent_open_files_max = 100

agent_restart_delay

Specifies the time, in minutes, when uxauthd restarts to recover from the critical problem.

Values: positive integer or -1 to cancel uxauthd restart.

Default: 60

Example: agent_restart_delay = 60

agent_vmemory_max

Specifies the maximum virtual memory size, in megabytes, that the UNAB agent can use. The UNAB agent restarts if it exceeds the maximum value.

Default: 300

Example: agent_vmemory_max = 300

debug_backup

Specifies whether to back up the debug messages file.

Limits: yes, no

Default: yes

debug_backup_file

Defines the name of the backup debug messages file. If you do not use a full pathname to the file, UNIX Authentication Broker creates the file in the directory InstallDir/log/debug/.

Default: agent_debug.back

debug_file

Defines the file name that UNIX Authentication Broker writes the debug messages to. If you do not use a full pathname to the file, UNIX Authentication Broker creates this file in the directory *InstallDir/log/debug/*.

Default: agent_debug

debug_size

Defines the maximum size of the debug messages file in megabytes.

Default: 512

Note: When the file exceeds the maximum size, the agent renames the file to backup and creates a messages file.

debug_level

Specifies the level of debug messages in the debug file.

Limits: disabled, high, medium, low

- disabled: Does not write debug messages to file
- high: Writes HIGH level debug messages to file
- medium: Writes HIGH and MEDIUM level debug messages to file
- low: Writes HIGH, MEDIUM and LOW debug message to file

Default: disabled

debug_zones

Specifies whether to log debug messages for submodules (zones). To write debug messages for more than one zone, specify the sum of the zone values.

Limits: -1, 1, 2, 4, 8, 16, or a sum of positive values.

- zone -1: Write debug messages for all zones.
- zone 1: Write debug messages for the General zone.
- zone 2: Write debug messages for the Entire communication zone.
- zone 4: Write debug messages for the Scheduler zone.
- zone 8: Write debug messages for the PAM communication zone.
- zone 16: Write debug messages for the NSS communication zone.

Example: To log debug messages for the zones "General" and "Scheduler", set the value of debug_zones to 5.

Default: -1

default_login_access

Specifies the default access mode if there are no rules that define access for users and groups.

Limits: 0 no access, 1 access granted

Default: 0

Note: This parameter is valid in full integration mode only.

groups_allow_file

Defines the location of the local groups.allow file.

Default: /opt/CA/uxauth/etc/groups.allow

Note: This parameter is valid in full integration mode only.

groups_deny_file

Defines the location of the local groups.deny file.

Default: /opt/CA/uxauth/etc/groups.deny

Note: This parameter is valid in full integration mode only.

heartbeat_send_interval

Defines the interval, in seconds, for sending a heartbeat to the CA Access Control Distribution Host.

Default: 3600

health_check_interval

Specifies the interval, in seconds, between the uxauthd internal self check.

Values: positive integer or -1 to cancel the self check.

Default: 300

Example: health_check_interval = 300

ldap_connection_lifetime

Defines the maximum period, in seconds, for keeping unused LDAP connections open. When set to 0 UNIX Authentication Broker destroys the connection immediately after an LDAP operation.

Default: 60

LIC98Dir

Defines the location of the CA license library.

Default: /opt/CA/SharedComponents/ca_lic

login_name_type

Specifies whether mapped users can log in using their UNIX user name or enterprise user name.

Limits: 1 - UNIX login name, 2 - enterprise login name

Default: 1

message_read_interval

Specifies the interval, in seconds, for reading the CA Access Control policy queue.

Default: 60

message_read_timeout

Defines the timeout period, in milliseconds, for reading the CA Access Control policy queue.

Default: 1

nss_cache_update_grp_login

Specifies whether NSS updates the group cache after every user login.

Limits: yes, no

Default: yes

Note: This parameter is valid in full integration mode only.

nss_cache_update_grp_mode

Specifies the group cache updating method.

Limits: 0 - no updating, 1 - incremental updating, 2 - full updating

Default: 1

Note: This parameter is valid in full integration mode only.

nss_cache_update_interval

Defines the interval, in minutes, for updating the users and groups cache.

Default: 60

Note: This parameter is valid in full integration mode only.

nss_cache_update_startup

Specifies the method of updating the NSS user and group cache during the agent startup.

Limits: 0 no updating, 1 incremental updating, 2 full updating

Default: 1

Note: This parameter is valid in full integration mode only.

nss_cache_update_usr_login

Specifies whether NSS updates the user cache after every user login.

Limits: yes, no

Default: yes

Note: This parameter is valid in full integration mode only.

nss_cache_update_usr_mode

Specifies the user cache updating method.

Limits: 0 no updating, 1 incremental updating, 2 full updating

Default: 1

Note: This parameter is valid in full integration mode only.

ntp_server

Defines the name or IP address of the NTP server.

Default: none

offline_logon

Specifies whether users can continue accessing the UNIX host when the Active Directory is not available.

Limits: no, offline connection disabled; yes, offline connection enabled

Default: yes

offline_logon_max_fail

Defines the maximum number of failed offline logon attempts.

Default: 5

offline_logon_period

Defines the maximum period, in days, that an offline authentication is permitted after the last successful online authentication.

Default: 30

report_user_mapped_name

Specifies the displayed user name in audit files and reports when the user is in mapped mode.

Limits: no, report displayed with the UNIX user name; yes, report displayed with the user mapped name.

Default: no

tgt_renew_interval

Defines the Ticket Granting Ticket (TGT) renewal interval in seconds.

Default: 7200

tgt_renewable_lifetime

Defines the Ticket Granting Ticket (TGT) renewal maximum period in days.

Default: 30d

time_sync_interval

Defines the clock synchronization interval in seconds.

Default: 300

unix_shells

Defines the rules for converting Active Directory users shell to a supported UNIX shell. If no match is found, then the shell that is defined as other is used.

Default (HP-UX):
sh=/sbin/sh,csh/sbin/csh,bash=/sbin/bash,ksh=/sbin/ksh,tcsh=/sbin/tcsh,other=/sbin/sh

Default (all other OS):
sh=/bin/sh,csh/bin/csh,bash=/bin/bash,ksh=/bin/ksh,tcsh=/bin/tcsh,other=/bin/sh

Note: This parameter is valid in full integration mode only.

use_local_policy

Specifies whether to use the local login policy (.allow and .deny files).

Limits: no, use the enterprise login policy only; yes, use the enterprise login policy and then the local login policy.

Default: no

use_nested_group_aclshell that is definedSpecifies whether nested groups are used for the user ACL.

Limits: no, nested groups are not used; yes, nested groups are used.

Default: yes

use_time_sync

Specifies the clocks synchronization options.

Limits: no, manual synchronization; yes, automatic synchronization

Default: no

use_wingrp

Specifies whether UNAB stores the Active Directory groups in a database for CA Access Control use.

To work in partial integration mode while CA Access Control is not integrated, disable group database creation when configuring UNAB.

Limits: no, yes

Default: yes

users_allow_file

Defines the location of the local users.allow file.

Default: /opt/CA/uxauth/etc/users.allow

Note: This parameter is valid in full integration mode only.

users_deny_file

Defines the location of the local users.deny file.

Default: /opt/CA/uxauth/etc/users.deny

Note: This parameter is valid in full integration mode only.

user_ticket_cleanup_interval

Specifies the cleanup interval, in seconds, for deleting expired user tickets.

Limits: any positive integer

Default: 3600

watchdog_check_interval

Specifies the time interval in which the UNAB watchdog checks for uxauthd existence.

Default: 60 seconds

Example: watchdog_check_interval = 60

watchdog_enabled

Specifies whether to use the UNAB watchdog to protect the daemon agent. The UNAB watchdog can also be run as a daemon when UNAB is installed without CA Access Control.

Default: yes

Example: watchdog_enabled = yes

wingrp_update_interval

Defines the interval, in minutes, for updating the UNIX Authentication Broker Active Directory groups database.

Default: 60

Note: This parameter is valid in full integration mode only.

wingrp_update_login

Specifies whether the Windows group database is updated every user login.

Limits: yes, no

Default: yes

Note: This parameter is valid in full integration mode only.

windgrp_update_mode

Specifies the method of updating the UNIX Authentication Broker Active Directory groups database.

Limits: 0 no updating, 1 incremental updating, 2 full updating

Default: 1

Note: This parameter is valid in full integration mode only.

wingrp_update_startup

Specifies the method of updating the Active Directory groups database during the UNIX Authentication Broker startup process.

Limits: 0 no updating, 1 incremental updating, 2 full updating

Default: 1

Note: This parameter is valid in full integration mode only.

working_threads

Defines the number of working threads in the agent.

Default: 64

global

The [global] section contains the parameters that control the UNIX Authentication Broker general settings.

activation

Specifies the host activation level.

Limits: 0, 1, 2

- 0 - not registered
- 1 - registered (login permitted for user defined in local user store only)
- 2 - activated (login is permitted for users defined in local user store or defined in either the .allow file or in the UNIX Authentication Broker login policy)

Default: 0

CASHCOMP

Specifies the path to the CA shared components install directory.

Default: /opt/CA/SharedComponents

integration_mode

Specifies the UNIX Authentication Broker installation method.

Limits: 1 - partial integration, 2 - full integration

Note: Specify partial integration (1) if you want to maintain the UNIX user store.

Default: 2

locale

Defines the language for the UNAB agent and utilities.

Example: C (English), japanese, chinese-s, chinese-t

Default: C

kerberos_configuration

Specifies how Kerberos configuration is used when implementing a UNIX Authentication Broker assisted Kerberos SSO.

Limits:

- internal—Specifies that the configuration file and user credential caches are stored under /opt/CA/uxauth and opt/CA/uxauth/etc directories
- external—Specifies that the configuration file and user credential caches are stored in their native locations

Note: This token is automatically configured during UNIX Authentication Broker registration.

Note: Linux, HPUX and Solaris store the user credentials in /tmp directory. AIX stores the user credentials in /var/krb5/security/creds directory

Default: internal

product_path

Defines the name of the UNIX Authentication Broker install directory.

Default: /opt/CA/uxauth

libdefaults

The [libdefaults] section contains tokens that control Kerberos configuration settings.

default_realm

Defines the default Kerberos realm for the UNIX Authentication Broker endpoint. A value of *unregistered* specifies that UNIX Authentication Broker does not use Kerberos.

Default: unregistered

dns_lookup_kdc

Specifies that UNIX Authentication Broker uses DNS SRV (service locator) records to look up the KDC (Key Distribution Centre) services location.

Limits: true, false

Default: true

dns_lookup_realm

Specifies that UNIX Authentication Broker uses DNS TXT records to look up domain to realm mappings.

Limits: true, false

Default: false

ticket_lifetime

Defines the ticket lifetime in seconds.

Default: 2400

logmgr

The [logmgr] section contains tokens that the UNIX Authentication Broker logging utility uses.

audit_back

Defines the full path name of the audit log backup file.

Default: /opt/CA/uxauth/log/uxauth.audit.bak

audit_group

Specifies the the name of the group that is permitted to read the audit log files.

Limits: none, *group_name*

- None - No group access, only root can read the audit log files
- *group_name* - Defines the name of the group that can read the audit log files

Note: If you change the value of this token after UNIX Authentication Broker creates the audit log file, you must use selang commands to set the file group ownership and the group permissions to read the log. Any files that are created after you set the value of this token will have the permissions that you specify.

Default: none

audit_log

Defines the full path name of the audit log file.

Default: /opt/CA/uxauth/log/uxauth.audit

audit_max_files

Defines the maximum number of audit log files to save for each of the specified backup modes. When the maximum number of backup audit log files is reached, UNIX Authentication Broker deletes the oldest backup file when it creates the newest. A value of 0 specifies that UNIX Authentication Broker keeps accumulating backup files.

Default: 0

audit_size

Defines the maximum size of the audit log file in KB.

Note: The minimum value you can specify for this token is 50 KB.

Default: 1024

audit_to_syslog

Specifies whether to log audit events to syslog file.

Limits: yes, no

Default: no

BackUp_Date

Specifies the interval for backing up the audit log files.

Limits: none, yes, daily, weekly, monthly

- none - performs the backup when the file reaches the size specified in the audit_size token but does not timestamp the file.
- yes - Audit log file backup is performed when the audit file reaches the size specified in the audit_size token
- daily - Audit log files backup is performed on a daily basis
- weekly - Audit log files backup is performed on a weekly basis

monthly - Audit log files backup is performed on a monthly basis

Note: If you specify daily, weekly, or monthly for this token, UNIX Authentication Broker creates a time stamp, backs up the audit log file when the current date surpasses the specified interval, and appends the time stamp to the name of the backup file. However, if the size of the audit log file reaches the size specified in the audit_size token before the current date surpasses the specified interval, UNIX Authentication Broker backs up the audit log file but does not append the time stamp to the name of the backup file. If you specify yes for this token, the time stamp is always appended to the name of the backup file.

Default: none

error_back

Defines the full path name of the error log file backup copy.

Default: /opt/CA/uxauth/log/uxauth.error.bak

error_group

Specifies the name of the group that is permitted to read the error log files.

Limits: none, *group_name*

- None - No group access, only root can read the error log files
- *group_name* - Defines the name of the group that can read the error log files

Note: If you change the value of this token after UNIX Authentication Broker creates the error log file, you must use selang commands to set the file group ownership and the group permissions to read the log. Any files that are created after you set the value of this token will have the permissions that you specify.

Default: none

error_log

Defines the full path name of the error log file.

Default: /opt/CA/uxauth/log/uxauth.error

error_size

Specifies the maximum size of the error log file in KB.

Note: The minimum value you can specify for this token is 50 KB.

Default: 50

map

Valid in Full Integration Mode

The [map] section contains tokens that UNIX Authentication Broker uses to specify Active Directory attribute names.

group_gid_attr_name

Specifies the Active Directory attribute name that indicates the UNIX group ID.

Default: gidNimber

group_member_attr_name

Specifies the Active Directory attribute name that lists members of a group.

Limits: member, memberUid

Note: Use value memberUid only when user_name_attr_name = msSFU30Name.

Default: member

user_gecos_attr_name

Specifies the Active Directory attribute name that indicates the UNIX user gecos.

Default: gecos

user_gid_attr_name

Specifies the Active Directory attribute name that indicates the UNIX group ID.

Default: gidNumber

user_homedir_attr_name

Specifies the Active Directory attribute name that indicates the UNIX user home directory.

Default: unixHomeDirectory

user_loginshell_attr_name

Specifies the Active Directory attribute name that indicates the UNIX user login shell.

Default: loginShell

user_name_attr_name

Specifies the Active Directory attribute name for the UNIX user name.

Default: sAMAccountName

user_uid_attr_name

Specifies the Active Directory attribute name that indicates the UNIX user ID.

Default: uidNumber

message

The [message] section contains tokens UNIX Authentication Broker uses to define the message file.

filename

Defines the full path name of the message file.

Default: /opt/CA/uxauth/data/uxauth.msg

migrate

The [migrate] section contains tokens that UNIX Authentication Broker uses during the migration process.

conflicts_file

Defines the full path name of the migration conflicts file.

Default: /opt/CA/uxauth/log/migrate.conflicts

create_ad_groups

Specifies whether to create new Active Directory groups during migration if no identical groups were found in Active Directory.

Limits: yes, no

Default: yes

disable_mapped_user

Specifies whether to disable the UNIX password of partially migrated (mapped) users.

Limits: yes, no

Default: yes

ignore_gecos_conflict

Defines whether to ignore gecos user attribute-related conflicts that UNIX Authentication Broker finds during the migration process.

Limits: yes, no

Default: yes

is_gid_migration_a_prerequisite

Specifies whether the migration of the user's primary group is a prerequisite to migrate the user.

Limits: yes, no

Default: no

journal

Defines the full path name of the migration journal file.

Default: /opt/CA/uxauth/log/migrate.journal

minimal_gid

Defines the minimal group ID that will be migrated to Active Directory during the migration process. Groups with a lesser GID are not migrated.

Default: 101

minimal_uid

Defines the minimal user ID that will be migrated to Active Directory during the migration process. Users with a lesser UID are not migrated.

Default: 101

remove_migrated_user

Specifies whether to remove the local user account after migration.

Limits: yes, no

Default: yes

try_to_map_on_conflict

Specifies whether to map conflicting accounts if the full migration process fails.

Limits: yes, no

Default: yes

passwd

The [passwd] section contains tokens that UNIX Authentication Broker uses to control password changes during the migration process.

YpGrpCmd

Defines the command to generate the NIS group map.

Default: make group

YpMakeDir

Defines the makefile directory that is used when creating NIS maps.

Default: /var/yp

YpPassCmd

Defines the command to generate the NIS password map.

Default: make passwd

YpServerGroup

Defines the full pathname of the group file on the NIS server.

Default: /etc/group

YpServerPasswd

Defines the full pathname of the password file on the NIS server.

Default: /etc/passwd

YpServerSecure

Defines the full pathname to the password file of the operating system.

Default (AIX): /etc/security/passwd

Default (HP-UX): /.secure/etc/passwd

Default (Solaris): /etc/shadow

Default (all other OS): /etc/shadow

pam

The [pam] section contains tokens that UNIX Authentication Broker uses to interact with the PAM module.

debug_mode_for_user

Defines whether the PAM module can print messages to the user screen during login.

Options: yes,no

Default: yes

home_directory_permission

Specifies the default file permissions that are assigned to users home directory

Values: 0-7

Default: 700

Example: 700—indicates that each user has write, read and execute permissions to their home directories only.

pam_exit_on_deny

Defines the PAM module behaviour if login was denied due to enterprise or local policy settings or Active Directory account state.

Options: yes;PAM module closes the sequence and prevent other PAM module from authenticating the user, no;PAM module does not close and enables other PAM modules to authenticate the user and permits the log in server to retry the PAM sequence call

Default: yes

pam_receive_timout

Specifies the time, in seconds, that the PAM module waits for the UNIX Authentication Broker agent (uxauthd) to respond.

Limits: any positive integer.

Default: 10

register

The [register] section contains tokens that control UNIX Authentication Broker registration functionality.

start_uxauthd

Specifies whether to run the uxactivate utility at the end of the installation process.

Limits: yes, no

Default: yes

verbose

Defines the verbose level to use during the installation process.

Default: 0

The UNIX Authentication Broker Conflicts File

The UNIX Authentication Broker conflicts file is created after you attempt to migrate users and group to Active Directory. The file details the conflicts that were discovered by UNIX Authentication Broker during the migration process. Review this file to resolve the conflicts that are reported in the file.

This file contains the following fields:

```
Solution Entity Type,Solution Entity Name,Solution Operation,Solution AD Mapped
Name,Conflicts,UID,Home Directory,GID,Member of,Members,GECOS
```

Solution Entity Type

Displays the type of solution entity to migrate.

Limits: user, group

Solution Entity Name

Displays the name of the entity.

Solution Operation

Displays the entity migration status.

Limits: Keeplocal, Migrate, Map

Solution AD Mapped Name

Displays the Active Directory account name that the local account is mapped to.

Conflicts

Displays the conflicts that were found during the migration.

UID

Displays the user ID.

Home Directory

Displays the user home directory.

GID

Displays the group ID.

Member Of

Displays the groups that the user is a member of.

Members

Displays a list of users that are members in the group.

GECOS

Displays GECOS information.

The SSH Device XML File

The SSH Device XML file lets you configure how Privileged User Password Management connects to an SSH Device endpoint, discovers user accounts, and changes privileged account passwords on the endpoint.

Different SSH Device XML files configure the interactions with different types of SSH Device endpoints. For example, the aix_connector_conf.xml file configures the connection to an AIX endpoint, and the device_connector_conf.xml file configures the connection to an SSH device such as a router.

Note: For more information about SSH Device XML file types, see the *Enterprise Administration Guide*.

The SSH Device XML files are located in the following directory:

ACServerInstallDir`/Connector Server/conf/override/sshdyn`

If required, you can customize the SSH Device XML files to suit your enterprise requirements.

Structure

The SSH Device XML file contains the following elements:

- <class name="SSHConnectionManager">—Contains parameters that manage the SSH connection
- <class name="CommandProcessor">—Contains parameters that specify connection settings
- <class name="CommandSet">—Contains the array elements that specify the commands that Privileged User Password Management executes on the endpoint

The <class name="CommandSet"> element contains array elements that group sets of commands, as follows:

- <array name="oGetUsers">—Contains the commands that Privileged User Password Management executes to get users
- <array name="oChangePassword">—Contains the commands that Privileged User Password Management executes to change user passwords
- <array name="oSubstituteUser">—Contains the commands that Privileged User Password Management executes to su to another user

  Note: The <array name="oSubstituteUser"> element is valid only for the aix_connector_conf.xml, checkpoint_connector_conf.xml, and ssh_connector_conf.xml files.

Each array element contains multiple <item> elements. An <item> element defines the parameters for a specific command that Privileged User Password Management executes on the endpoint. For example, an <item> element in the <array name="oGetUsers"> element may specify the:

- Command that Privileged User Password Management executes to get local users
- Length of time that Privileged User Password Management waits for a response
- Text string for which Privileged User Password Management waits to receive before proceeding
- Text string in the response that indicates the command failed

Note: For examples of how <item> elements in the SSH Device XML file configure interactions with SSH Device endpoints, see the *Enterprise Administration Guide*.

You use nested parameters to define the configuration settings for each element, as follows:

- The <class name="SSHConnectionManager"> and <class name="CommandProcessor"> elements contain parameters that define connection settings
- <item> elements contain parameters that define the parameters for a specific command

Each nested parameter has the following format:

```
<param name="name" value="value" />
```

The following snippet of an SSH Device XML file shows how the elements are nested:

```
<package name="com.ca.jcs.sshdyn">
        <class name="SSHConnectionManager">
                <param name="name" value="value" />
        </class>
</package>
<package name="com.ca.sessame.conn.unix">
        <class name="CommandProcessor">
                <param name="name" value="value" />
        </class>
        <class name="CommandSet">
                <instance name="ssh">
                        <array name="oGetUsers">
                                <item>
                                        <param name="name" value="value" />
                                </item>
                        </array>
                        <array name="oChangePassword">
                                <item>
                                        <param name="name" value="value" />
                                </item>
                        </array>
                        <array name="oSubstituteUser">
                                <item>
                                        <param name="name" value="value" />
                                </item>
                        </array>
                </instance>
        </class>
</package>
```

Elements

SSHConnectionManager

Specifies the settings that Privileged User Password Management uses to manage the SSH connection.

This class element contains the following parameter:

I_CONNECTIONS

Defines the number of concurrent connections to the endpoints.

Default: 10

CommandProcessor

Specifies the settings that Privileged User Password Management uses to connect to the SSH Device endpoint.

This class element contains the following parameters:

bToLog

Specifies whether Privileged User Password Management writes messages to sLogFileName.

Limits: true, false

Default: true

sLogFileName

Defines the relative pathname to the log file.

Default: ..\logs\uxlog.txt

limitResultCharsToLog

Defines the maximum number of characters CA Access Control writes to the log file for each connection.

Default: 1500

bSkipOperationAdminTestConnection

Specifies

Limits: true, false

Default: true

maxTimeLimit

Defines the maximum time, in milliseconds, that Privileged User Password Management waits for values.

Default: 1500

waitIntervalDefault

Defines the time, in milliseconds, that Privileged User Password Management wait

Default: 500

login_str

Specifies the Telnet request command for a user name.

Example: login

password_str

Specifies the Telnet request command for password.

Example: password

AYT_answer

Specifies the answer that the device for the Telnet command "Are You There"

Default: Solaris-Yes, Linux-yes, AIX-here

Note: Due to different configurations, each SSH device can have a unique reply to the AYT command. You can modify the SSH XML file accordingly.

To discover the format, open a telnet session to the device and run the following:

```
^+]
send ayt
```

iPort

Defines the SSH port number.

Note: By default, this parameter is commented out.

Default: 22

CommandSet

Specifies the commands that Privileged User Password Management executes on the endpoint.

This class element contains array elements that group the commands that Privileged User Password Management executes on the endpoint.

oGetUsers

Specifies the commands that Privileged User Password Management executes to get users.

This array element contains item elements that define the parameters for the specific commands that Privileged User Password Management executes to get users.

oChangePassword

Specifies the commands that Privileged User Password Management executes to change user passwords.

This array element contains item elements that define the parameters for the specific commands that Privileged User Password Management executes to change user passwords.

oSubstituteUser

Specifies the commands that Privileged User Password Management executes to su to another user.

This array element contains item elements that define the parameters for the specific commands that Privileged User Password Management executes to su to another user.

Note: This element is valid only for the aix_connector_conf.xml, checkpoint_connector_conf.xml, and ssh_connector_conf.xml files.

item

Specifies the parameters for a specific command that Privileged User Password Management executes on the endpoint.

Each item element may contain the following parameters:

sCommand

Defines the command that Privileged User Password Management sends to the endpoint.

iWait

Defines the interval, in milliseconds, that Privileged User Password Management waits until it performs the next step.

Default: 500

sWaitForText

Defines the text string that Privileged User Password Management waits to receive in response to the command defined in sCommand.

sFailureResult

Defines the text string that Privileged User Password Management receives from the endpoint that indicates the command failed.

sToFilterOut

Defines the text strings that Privileged User Password Management removes from the endpoint output.

bHideSentLog

Specifies whether to write commands to the log file.

Limits: true - Privileged User Password Management does not write commands to the log file, false - Privileged User Password Management does write commands to the log file

Default: true

sTrueResultRegex

(Optional) Specifies to compare the command results with the specified string. If the result does not match the string, an error message is displayed.

Note: By default, this parameter is commented out.

iXMLVersion

Indicates the XML file version. The XML version cannot be later than the XML version that is defined in the SSL connector.

Default: 0

ToReport

Specifies whether XML processing data is logged to $XML_NAME..lodaing_report.xml. The log file is located in the following directory:

ACServerInstallDir/Connector Server/conf/override/sshdyn

Limits: true, false

Default: true

FileIsLoaded

Indicates that the XML file was loaded successfully.

Default: OK

The Privileged User Password Management Automatic Login Application Visual Basic Script

The Privileged User Password Management automatic login application uses Visual Basic scripts to enable automatic users login. You can customize the Visual Basic scripts to create new login applications or modify existing login applications.

The Privileged User Password Management automatic login application script contains variables that the ActiveX replaces with values when downloaded to the client machine from the Enterprise Management Server. The Enterprise Management Server processes the scripts and replaces the keywords with values. The ActiveX then executes the script on the client machine.

The Privileged User Password Management automatic login application scripts are located in the following directory:

JBOSS_HOME/server/default/deploy/IdentityMinder.ear/config/sso_scripts

Elements

The Privileged User Password Management login application script contains the following keys:

#host#

Specifies the name of the endpoint that the user automatically logs in to

#username#

Specifies the checked out privileged account

#password#

Specifies the privileged account password to check out

#userdomain#

(Active Directory) Specifies the privileged account domain name

#isActiveServletUrl#

Specifies the URL that the ACLauncher ActiveX uses to check for an account password check in event.

#CheckinUrl#

Specifies the URL that the ACLauncher ActiveX uses to check in the account password in case the user logged out of the endpoint.

#SessionidUrl#

Specifies the URL that the ACLauncher ActiveX uses to send recorded session ID if the sessions is recorded in ObserverIT Enterprise

The following snippet of a Privileged User Password Management automatic login application script displays how the variables appears:

```
Set pupmObj = CreateObject("ACLauncher.ACWebLauncher")
hwnd = pupmObj.LauncheRDP("#host#", "#userDomain#\#userName#", "#password#")
' Set window close event
pupmObj.SetWindowCloseEvent(hwnd)
' Set server checkin event
pupmObj.SetServerCheckinEvent("#isActiveServletUrl#")
' Wait until one of the events signaled
rc = pupmObj.WaitForEvents()
If rc = 1 Then 'user has closed the window - notify the server side
        pupmObj.SendCheckinEvent("#CheckinUrl#")
ElseIf rc = 2 Then 'timeout elapsed - close the window
    call pupmObj.CloseWindow(hwnd, 0)
ElseIf rc = 3 Then 'the account was checkedin at the server side - close the window
    call pupmObj.CloseWindow(hwnd, 120)
End If
```

Structure

The Privileged User Password Management automatic login application script structure is as follows:

- Initialization of the COM object

  ```
  Set pupmObj = CreateObject("ACLauncher.ACWebLauncher")
  ```

- Execution of the automatic login application

  ```
  hwnd = pupmObj.LauncheRDP("#host#", "#userDomain#\#userName#", "#password#")
  ```

- Post execution tasks—password check in, interactive login or timeout

  ```
  ' Wait until one of the events signaled
  rc = pupmObj.WaitForEvents()
  If rc = 1 Then 'user has closed the window - notify the server side
      pupmObj.SendCheckinEvent("#CheckinUrl#")
  ElseIf rc = 2 Then 'timeout elapsed - close the window
      call pupmObj.CloseWindow(hwnd, 0)
  ElseIf rc = 3 Then 'the account was checkedin at the server side - close the window
      call pupmObj.CloseWindow(hwnd, 120)
  End If
  ```

To record the login application session, add recording instructions to the script, as follows:

- In the initialization section. add the following:

```
Set observeIT = CreateObject("ObserverIT.AgentAPI.Proxy")
```

- In the application execution section, add the following:

```
'Get application processid
processID = pupmObj.GetWindowProcessID(hwnd)
'Start recording
sessionid = observeIT.StartByProcessID(processID, true)
'Send the sessions if to the ENTM server
pupmObj.AssignSessionID "#SessionidUrl#" ,sessionId
```

- In the post execution section, add the following:

```
'Stop recording

observeIT.StopBySessionId sessionId, true
```

Methods

The ACLauncher ActiveX uses the following methods:

```
LauncheRDP (BSTR bsHostName, BSTR bsUserName, BSTR bsPassword, VARIANT *phWindow);
```

Launch the remote desktop session with the input credentials and return the remote desktop window handle

Example: Dim test Set test = CreateObject("ACLauncher.ACWebLauncher") Hwnd = test.LauncheRDP("hostname.com", "hostname\administrator", "password")

```
LaunchePUTTY (BSTR bsHostName, BSTR bsUserName, BSTR bsPassword, VARIANT *phWindow);
```

Launch the PuTTY session with the input credentials and return the PuTTY window handle

Example: Dim test Set test = CreateObject("ACLauncher.ACWebLauncher") Hwnd = test. LaunchePUTTY ("hostname.ca.com", "root", "password")

```
LauncheProcessAsUser (BSTR bsApplication, BSTR bsCommandline, BSTR bsUsername, BSTR bsPassword, VARIANT *phWindow);
```

Launch process with the input credentials and return the process window handle

Example: Dim test Set test = CreateObject("ACLauncher.ACWebLauncher") Hwnd = test.LauncheProcessAsUser("cmd.exe", "/k echo This console is run under %USERNAME% account...", "administrator" , "password")

```
GetWindowProcessID(VARIANT *phWindow, LONG *pProcessID);
```

Return the process ID of a specified window handle

Example: Set test = CreateObject("ACLauncher.ACWebLauncher") hwnd = test.LauncheRDP("hostname", "administrator", "password") id = test.GetWindowProcessID(hwnd) test.Echo "Process ID = " & id

```
GetWindowTitle(VARIANT *phWindow, BSTR *pbsTitle);
```

Return the Title of a specified window handle

Example: Set test = CreateObject("ACLauncher.ACWebLauncher") hwnd = test.LauncheRDP("hostname", "administrator", "password") title = test.GetWindowTitle(hwnd)

```
CloseWindow(VARIANT *phWindow, LONG Seconds);
```

Display a dialog box with a message specifying that the window will close in X seconds and close the window of a specified window handle

Example: Set test = CreateObject("ACLauncher.ACWebLauncher") hwnd = test.LauncheRDP("hostname", "administrator", "password") test.Sleep(5000) test.CloseWindow(hwnd, 60)

```
SetTimeoutEvent(LONG seconds);
```

Specify the timeout for "WaitForEvents" method. Once reached, the WaitForEvents method returns from its blocking call with a return value that indicates the timeout reached

Example: Set test = CreateObject("ACLauncher.ACWebLauncher") hwnd = test.LauncheRDP("hostname", "administrator", "password") test.SetTimeoutEvent(10)

```
SetWindowCloseEvent(VARIANT *phWindow);
```

Specify the window closing event for the "WaitForEvents" method. After the window is closed, the "WaitForEvents" method returns from its blocking call and displays the return value that indicates that the window was closed

Example: Set test = CreateObject("ACLauncher.ACWebLauncher") hwnd = test.LauncheRDP("hostname", "administrator", "password") test.SetWindowCloseEvent(hwnd)

```
SetServerCheckinEvent(BSTR bsURL);
```

Sets the Privileged User Password Management check in event as a block execution condition. The ActiveX queries Privileged User Password Management every 5 seconds

Example: Set test = CreateObject("ACLauncher.ACWebLauncher") hwnd = test.LauncheRDP("hostname", "administrator", "password") test.SetServerCheckinEvent("http://server.com/__azy?djfhwek5jy34brfhwkeb") (replace with variable)

```
WaitForEvents(VARIANT *pRetVal);
```

Blocks the script execution until one of the register conditions is correct.

Options:1—the user closed the window, 2—timeout elapsed, 3—password checked in at the server side

Example: Set test = CreateObject("ACLauncher.ACWebLauncher") hwnd = test.LauncheRDP("hostname", "administrator", "password") test.SetServerCheckinEvent("http://server.com/__azy?djfhwek5jy34brfhwkeb")

test.SetWindowCloseEvent(hwnd) test.SetTimeoutEvent(360) rc = test.WaitForEvents() If rc = 3 Then call test.CloseWindow(hwnd, 10) End If

```
SwitchToThisWindow(VARIANT *phWindow);
```

Positions the window at the top of the Z order

Example: Set test = CreateObject("ACLauncher.ACWebLauncher") hwnd = test.LauncheRDP("hostname", "administrator", "password") test.SwitchToThisWindow(hwnd)

```
SendCheckinEvent(BSTR bsURL);
```

Send check in event when user closes the window

Example: Set test = CreateObject("ACLauncher.ACWebLauncher") hwnd = test.LauncheRDP("hostname", "administrator", "password")

```
Sleep(LONG milliseconds);
```

Pauses the script execution

Set test = CreateObject("ACLauncher.ACWebLauncher") hwnd = test.Sleep(2000)

```
Echo(VARIANT* pArgs);
```

Print messages to screen

Set test = CreateObject("ACLauncher.ACWebLauncher") hwnd = test.Echo("Password Checkin")

Chapter 4: Registry Entries

This section contains the following topics:

The CA Access Control Registry

CA Access Control creates its registry entries under the following registry key, which is called ACROOT in CA Access Control Endpoint Management Remote Configuration:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl
```

The main registry key contains the following registry entries:

CurrentVersion

Defines the current version and build of product.

Encryption Package

Defines the full path name of the DLL used to implement symmetric encryption.

Default: *ACInstallDir*\bin\aes256enc.dll

<Build_Number>

CA Access Control defines the current version and build of product in the following registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Build_Number
```

This key is for internal use only.

AccessControl

CA Access Control maintains generic settings it uses under the following key:

`HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\AccessControl`

The AccessControl registry key contains the following registry entries:

AccessControl Services

Defines a list of CA Access Control service names and the executable.

Default: "SeOSAgent;SeOS Agent", "SeSudo;SeOS TD", "seoswd;SeOS Watchdog"

Note: The endpoint that is part of the Enterprise Management Server also contains the following default values for this registry entry: "Sepmdd;SeOS Policy Model(DMS__)", "Sepmdd;SeOS Policy Model(DH__)", "Sepmdd;SeOS Policy Model(DH__WRITER)"

admin_default_check

Specifies whether CA Access Control is denied login access to the CA Access Control server, even when the *defaccess* property for a remote terminal resource is set to *all*, or access to _default terminal resource is permitted.

Maintained for backward compatibility.

Default: 0 (access is not denied)

AdminInst

Internal use only.

Default: 0

auth_login

Specifies how a user is authenticated for administration purposes.

Valid values are:

native - for native operating system users, checks the user password against OS.

eTrust - for users that do not exist in the native operating system, checks the user password against CA Access Control database.

Default: native

auth_module_names

The list of language client modules that are allowed to authenticate outside of native authentication. Client module name is set by the client inside the lca API calls before the authentication. Changing this registry value may affect other clients authenticating in a non native mode.

Default: none

CPF_TARGETS

List of target mainframe CPF systems (remote CPF target nodes) that the CPF service communicates with.

Default: ACF2 TOP RACF

eACPipePrefix

A value for part of the pipe name that the new pipe servers and pipe clients will use. If a system has older clients of CA Access Control, then this value is obligatory for those clients to work. Otherwise, change this value to a more secure pipe name.

Default: SEOS

eACPipeTranslator

Obsolete.

full_year

Specifies whether years appear in two-digit (value=no) or four-digit (value=yes) format, when using the secons -tv, seaudit, and dbmgr utilities.

Default: yes

GenerateMemDump

Specifies whether CA Access Control creates a memory dump (1) when handling a code exception of a CA Access Control service. CA Access Control creates the memory dump in *ACInstallDir*\bin*serviceProcessName*.*PID*.dmp For example, SeOSAgent.5704.dmp

Note: The memory dump is only for user mode and not kernel mode.

Default: 1

parent_pmd

The PMDB to which this workstation subscribes in the format of *pmdb@host*. This is the only policy model that can update the local database.

If you do not specify a value, the workstation does not accept updates from any PMDB. If you set the entry to _NO_MASTER_, then any PMDB can update this workstation

No default.

Example: pmd1@host1;pmd2@host1;pmd3@host2

passwd_pmd

The target for password replacement on the policy model in the format *pmdb@host*.

The parent_pmd and passwd_pmd registry values can have the same value. If the parent_pmd and passwd_pmd registry values are not the same, the passwd_pmd database sends its updates to the parent_pmd database for distribution. The parent_pmd database must be a subscriber of the passwd_pmd database.

If you do not set this value, it inherits the value of the parent_pmd registry key.

No default.

ReverseIpLookup

Controls the way the client IP address is resolved to determine whether the user is allowed to log in from that terminal.

Valid values are:

yes-looks up the IP address of the open client's socket and logon is permitted accordingly.

no-uses the host name as received from the client and does not resolve any host names. (The same effect can be achieved by disabling class TERMINAL.)

Default: yes

secondary_pmd

The policy model database used as the secondary target for password replacement

No default.

SeOSPath

The directory in which CA Access Control is installed.

SplashEnable

The toggle to enable or disable a protection message during interactive (GINA) login process. This message tells the user that CA Access Control protects the computer. A value of 1 indicates the message is enabled; a value is 0 indicates that it is disabled.

Default: 1

TNG_Environment

The toggle to enable or disable Unicenter integration.

Values: 1—Enable Unicenter integration and create the database with the Unicenter TNG classes, 0—disable Unicenter integration and create the database without the Unicenter TNG classes

Default: 0

TrustedServices

List of trusted programs.

No default.

UseFsiDrv

Toggle to enable or disable driver loading.

Values: 1—Enable driver loading, 0—disable driver loading

Default: 1

Agent

CA Access Control maintains agent settings it uses under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Agent
```

Agent key entries (and any subkeys) are for internal use only.

ShutdownWaitingTimeout

Defines the timeout period, in milliseconds, the CA Access Control Agent waits for its components to gracefully shut down. If CA Access Control components do not shut down gracefully, the Agent shuts down forcefully.

Note: This registry entry is for internal use only.

Default: 60000

Applications

CA Access Control maintains application settings it uses under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Applications
```

The Applications registry key contains the following registry entries:

OperationMode

Specifies whether the controlled application mode is active (1).

This value has to be set to 1.

Default: 1

<Application_Name>

CA Access Control maintains specific application settings it uses under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Applications*Application_Name*

Each Applications*Application_Name* registry key contains the following registry entries:

ApplicationName

Defines the name of controlled process.

You must specify the full pathname in this format: *device*:*path**name*.exe.

Default: Full pathname to executable

Arguments

Defines arguments CA Access Control uses when starting the application.

Default: "" (no arguments)

Desktop

Defines the workstation and session name.

Default: No default

OperationMode

Specifies whether the application is active (1).

Default: 1

RestartApplication

Specifies whether the application will be restarted (1) if it has been closed or terminated.

Default: 1

StartAplication

Specifies whether CA Access Control be starts the application (1) when the Watchdog wakes up.

Default: 1

WorkingDirectory

Defines the working directory in which the application is started.

Default: *ACInstallDir*\bin

Client

CA Access Control maintains client application settings it uses under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Client

The Client registry key contains the following registry entries:

ConnectTo

Defines the host name CA Access Control client administration applications (for example, selang) connect to by default.

Default: localhost

Standalone

CA Access Control maintains standalone client settings it uses under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Client\Standalone

The Client\Standalone registry key contains the following registry entries:

full_login_check

The toggle to enable the CA Access Control server to check additional user properties (grace and max_login) and perform a login during a connection request from a standalone application.

This value helps remote password changes if one is about to expire.

If the value is set to 1, the checks are enabled.

Default: 0

Common

CA Access Control maintains settings used by common components under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Common

The Common key does not contain any registry entries. It contains registry subkeys for common components.

AgentManager

CA Access Control maintains the Agent Manager related settings in the following location:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\common\AgentManager
```

The Agent Manager registry key contains the following registry entries:

RefreshTimeout

Defines the Agent Manager refresh interval, in seconds.

Type: REG_DWORD

Default: 600

StandAloneService

Specifies whether or not this service is a standlone service.

Type: REG_DWORD

Default: 0

TraceEnabled

Defines the CA Access Control Agent Manager trace mode.

Options: 0,1

Default: 1

WorkSpace

Specifies the full pathname of the CA Access Control Agent Manager workspace.

Default:\ProgramFiles\CA\AccessControlShared\APMS\AccessControl\Data\Agent Manager

Plugins

CA Access Control maintains settings used by the plugins under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Common\AgentManager\
Plugins
```

The Plugins key does not contain any registry entries. It contains registry subkeys for plugins.

AccountManager

CA Access Control maintains the Account Manager related settings in the following location:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\common\AgentManager\
Plugins\AccountManager
```

The Account Manager registry key contains the following registry entries:

Interval

Defines the plugin schedule, in seconds.

Default: 1

Note: Applicable only when ScheduleType is set to 2.

OperationMode

Defines the plugin operation mode.

Options: 0 - plugin disabled, 1 - plugin enabled

Default: 1

PluginPath

Defines the full pathname of the plugin.

Type: REG_SZ

Default:
\ProgramFiles\CA\AccessControlServer\APMS\AccessControl\bin\AccountManager.dll

QueryFilter

Specifies additional values that is added to the Message Queue receive queue filter.

Options: ENDPOINT_CUSTOM 1...5=, ENDPOINT_OWNER=, ENDPOINT_DEPARTMENT=

Note the following:

- Place property values in apostrophes
- Use AND and OR operands to specify more that a single property
- Use parenthesis when needed

Schedule

Defines the plugin scheduling string.

Default: 00:00@Sun.Mon,Tue,Wed,Thu,Fri,Sat

Note: Applicable only when ScheduleType is set to 2.

ScheduleType

Define the plugin schedule type.

Options: 0 -execute once, 1 - execute on demand, 2 - execute on interval, 3 - execute on schedule

Default: 1

communication

CA Access Control maintains the message queue server communication settings it uses under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\common\communication
```

The communication registry key contains the following registry entries:

certificate

Defines the certificate file for the SSL connection.

Limits: The full pathname to a file containing the certificate data.

Distribution_Server

Defines the Distribution Server URL. You can define more than one Distribution Server in a comma-separated list.

Example: tcp://ds.comp.com:7222, tcp://ds_dr.comp.com:7222

Default: none

endpoint_to_server_queue

Defines the name of the message queue that the endpoint uses to send information to CA Access Control Enterprise Management.

Default: ac_endpoint_to_server

server_to_endpoint_broadcast_queue

Defines the name of the message queue that CA Access Control Enterprise Management uses to broadcast messages to all endpoints.

Default: ac_server_to_endpoint_broadcast

server_to_endpoint_queue

Defines the name of the message queue that CA Access Control Enterprise Management uses to send messages to the endpoint.

Default: ac_server_to_endpoint

ssl_custom

Specifies whether to use the host name verifier function.

Limits: 0, do not use the host name verifier function; 1, use the host name verifier function

Default: 0

ssl_hostname

Defines the SSL host name.

Default: none

ssl_identity

Defines the identity of the Report Agent.

Limits: The full pathname to a file containing the certificate data.

Default: none

ssl_issuer

Defines issuer certificates to the SSL connection.

Limits: The full pathname to a file containing the certificate data.

Default: none

ssl_key

Defines the Report Agent private key.

Limits: The full pathname to a file containing the private key.

Default: none

ssl_noverifyhost

Specifies whether to enable verification of the host certificate.

Limits: 0, disable host certificate verification; 1, enable host certificate verification

Default: 0

ssl_noverifyhostname

Specifies whether to enable verification of the host name.

Limits: 0, disable host name verification; 1, enable host name verification

Default: 0

ssl_trace

Specifies whether to enable SSL tracing.

Limits: 0, disable SSL tracing; 1, enable SSL tracing

Default: 0

ssl_trusted

Defines trusted certificates to the SSL connection.

Limits: The full pathname to a file containing the certificate data.

Default: none

crypto

CA Access Control maintains cryptography module settings it uses under the following key:

`HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\crypto`

The crypto registry key contains the following registry entries:

ca_certificate

Defines the full pathname to the Certificate Authority (CA) certificate database.

Default: *ACInstallDir*\data\crypto\def_root.pem

communication_mode

Specifies whether secure socket layer (SSL) protocols are enabled.

If you set this to ssl_only, only SSL V2, SSL V3, and TLS connections are enabled. This means that this computer cannot communicate with computers that do not support SSL, and so cannot communicate with computers that are running versions of CA Access Control earlier than r12.0, which do not support SSL.

Note: Computers that are running CA Access Control r12.0 and later do support SSL.

If the fips_only token is set to 1, the actual communication mode is set to ssl_only in FIPS mode (that is, TLS), and the communication_mode token is ignored.

Valid values are:

- all_modes
- ssl_only
- non_ssl

Default: non_ssl

encryption_methods

Specifies the encryption libraries that the CA Access Control Agent uses to decrypt messages. The Agent attempts to use each library in the list, in turn, until the decryption is successful.

Limits: aes256enc, aes192enc, aes128enc, desenc, tripledesenc, defenc

Default: aes256enc, aes192enc, aes128enc, desenc, tripledesenc

fips_only

This token controls whether CA Access Control works in FIPS only mode. In this mode all non-FIPS functions are disabled.

Valid values:

1 CA Access Control works in FIPS only mode

0 CA Access Control works in non-FIPS mode

Default: 0

private_key

Defines the full pathname to the subject private key.

Default: *ACInstallDir*\data\crypto\sub.key

ssl_port

Defines the port for SSL communications between CA Access Control clients and services.

Default: 5249

subject_certificate

Defines the full pathname to the subject certificate.

Default: *ACInstallDir*\data\crypto\sub.pem

Data

CA Access Control maintains internal settings it uses under the following key:

`HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Data`

Data key entries are for internal use only. You cannot open this key.

Dependency

CA Access Control maintains dependency settings it uses under the following key:

`HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Dependency`

When the CA Access Control component module is installed as an embedded component of another product, all subkeys of this registry key are the name of the product that is dependent on CA Access Control. If you upgrade or uninstall CA Access Control, CA Access Control checks this registry and decides whether the process can continue or if it must be aborted.

devcalc

CA Access Control maintains deviation calculator settings it uses under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\devcalc
```

The devcalc registry key contains the following registry entries:

dms_cmd_retry_interval

Defines the number of seconds between each DMS notification command retry.

Default: 60

max_dms_cmd_retry

Defines the maximum number of times the policy deviation calculator retries to send update notifications to the DMS before giving up.

Default: 3

max_lines_request

Defines the maximum number of lines (from the policy deviation data file) that the *get devcalc* selang command returns at any one time. You then need to retrieve additional lines using the following command:

```
get devcalc params("offset=X")
```

X

Defines the line offset returned by the previous *get devcalc* output.

Default: 50

Exits

CA Access Control maintains agent exit settings it uses under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Exits
```

The Exits registry key does not contain any registry entries. It contains registry subkeys for every agent exit.

AuthenticatePassword

CA Access Control maintains password authentication agent exit settings it uses under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Exits\AuthenticatePassword
```

The Exits\AuthenticatePassword registry key contains the following registry entries:

Enable

The toggle to enable or disable the password rules enforcement agent exit. A value of 0 disables the exit. Any other value enables it.

Default: 0

EnforcePasswordControl

The conditions for password rules enforcement using a CA Access Control client:

0 - no password rules enforcement

1 - password rules enforcement is activated when regular users change their own passwords

2 - password rules enforcement is activated when an admin or a password manager changes someone else's or their own password

3 - accumulation of values 1 and 2

Default: 1

Engine

CA Access Control maintains CA Access Control engine (seos) agent exit settings it uses under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Exits\Engine
```

The Exits\Engine registry key does not contain any registry entries by default.

Remote Grace Info

CA Access Control maintains remote grace information agent exit settings it uses under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Exits\Remote Grace Info
```

The Exits\Remote Grace Info registry key contains the following registry entry:

DefaultWarningDays

Defines the default number of days for a password expiration warning display to users of segrace\SegraceW utilities. It means that if one of these utilities is being applied and the password of the user is to expire in fewer days than specified by this registry value, then a warning message for the user is displayed.

Default: 7

Remote Shutdown

CA Access Control maintains remote shutdown agent exit settings it uses under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Exits\Remote Shutdown
```

The Exits\Remote Shutdown registry key contains the following registry entries:

Path

The full path name of the remote shutdown DLL.

Default: *ACInstallDir*\bin\remshut.dll

Prefix

The defined prefix used by the remote shutdown DLL.

Default: SD

FsiDrv

CA Access Control maintains driver settings it uses under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\FsiDrv
```

The FsiDrv registry key contains the following registry entries:

AuditRefreshPeriod

Defines the minimum time in seconds between two consecutive audit events from the same source. CA Access Control does not log audit messages for consecutive events from the same source that occur within this time period.

Default: 0 (all audit events are logged)

BatchOplockStatus

Specifies whether to disable batch OpLocks (opportunistic locking) of an entire file. When disabled (value is zero), the driver collects 100 percent of audit information for file access but performance decreases. A non-zero value keeps batch OpLocks operating regularly (enabled) and increases performance, but potentially provides incomplete audit information that may not include attempts to access related files.

Note: You must reload the driver to use the new setting. Unload the driver (net stop seosdrv) after you stop CA Access Control (secons -s).

Default: 1 (enabled)

CacheLimit

Defines the seosdrv kernel memory cache limit size in megabytes.

Type: REG_DWORD

Limits: 8 - 64

Default: 16

directory

The location of the driver.

Default: *system_drive**Windows_path*\system32\drivers

DynamicSysThreadDetection

Specifies that CA Access Control traces all kernel mode threads that are created by another product which creates system threads, for example Trend Micro™ PC-cillin Antivirus.

Note: Enabling this registry value can cause performance issues. We recommend that you contact CA Technologies before you enable this registry value. For assistance, contact CA Support at http://ca.com/support.

Type: REG_DWORD

Default: 0 (disabled)

FileCacheDisabled

The toggle to enable or disable the generic file cache.

Values: 0—enable the generic file cache, 1—disable the generic file cache

Default: 0

LoopHoleProtectionDisabled

Specifies whether to disable loophole protection, which protects CA Access Control from applications such as Process Monitor (procmon.exe) that may close its handles.

Values: 0 - enable loophole protection; **1** - disable loophole protection.

Default: 0

Note: This key applies to 32-bit Windows environments.

MaxAuditRecordLimit

Defines the audit queue limit. When the queue length exceeds this limit, CA Access Control artificially slows down threads that generate audit events so that it can read the queue and write to the log file faster than new items are added to the queue.

Note: When new items are added to the queue faster than CA Access Control can read and process them, the system's memory may be exhausted.

Default: 200

MaxTimeoutLimit

Defines the number of consecutive timeouts that CA Access Control detects before it triggers a driver bypass. Once reached, the driver stops sending authorization requests to the authorization engine until the engine indicates that it is ready to process events.

A value of zero disables this bypass.

Default: 5

NetworkDispatchLevelAccess

Defines the driver response during intercepted network event at dispatch at IRQL.

Values: 0,1

Default:

QueueTimeoutatch

The maximum time in seconds to wait for seosd to respond.

Default: 10

QueueTimeoutAnswer

The driver's response after time-out.

Default: 0 (Deny)

RegistryCacheDisabled

The toggle to enable or disable the generic registry cache.

Values: 0—enable the generic registry cache, 1—disable the generic registry cache

Default: 0

SilentModeAdmins

Line separated list of user names who can administer the computer in maintenance mode (SilentModeEnabled =1).

No default

SilentModeEnabled

Determines whether maintenance mode is active (1).

Default: 0 (disabled)

SystemBypassRestricted

Specifies if CA Access Control bypasses access checks for system processes. By default, CA Access Control does not consider system processes to be trusted and does not bypass access checks for system processes.

Values: 0 - bypass access checks; **1** - do not bypass access checks.

Default: 1

Instrumentation

CA Access Control maintains cainstrm.dll behavior settings (which apply to all loaded plug-ins) it uses under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Instrumentation
```

The Instrumentation registry key contains the following registry entries:

Active

Specifies whether cainstrm.dll is active (1).

If you specify 0, cainstrm.dll loads but does not process any plug-ins.

Type: REG_DWORD

Default: 1

ApplyOnProcess

Defines the list of processes to which instrumentation applies.

You can define the name of the service or the full pathname. Names are *not* case sensitive. For example, "services.exe", "\system32\services.exe", "c:\windows\system32\services.exe".

Type: REG_MULTI_SZ

By default, this token is not set (instrumentation applies to any process).

ExcludeProcess

Defines the list of processes to which instrumentation does *not* apply.

Note: This entry is valid only if ApplyOnProcess is not set.

Type: REG_MULTI_SZ

By default, this token is not set.

OperationMode

Specifies whether cainstrm.dll loads plug-ins (1) into memory.

Type: REG_DWORD

Default: 1

RunTimeInstrumentationDisabled

Specifies the CA Access Control instrumentation policy at run time.

Type: REG_DWORD

Limits: 0, runtime instrumentation enabled; 1, runtime instrumentation disabled.

Default: 0

RunTimeInstrumentationIncludeList

Defines a list of processes to apply the runtime instrumentation to.

Type: REG_MULTI_SZ

Default: empty

TraceDbgEnable

Specifies whether to trace status flag for the cainstrm module, that is, enables tracing into DbgView or Kernel Debugger.

Type: RED_DWORD

Limits: 0, false; 1, true.

Default: 0

TraceFileIsCyclic

Specifies the type of the trace file.

Type: REG_DWORD

Limits: 0, trace file is not cyclic; 1, trace file is cyclic.

Default: 0

TraceFileSizeLimit

Defines the maximum size of the trace file in bytes. A value of 0 means no maximum size limit is imposed on the trace file.

Type: REG_DWORD

Default: 0

TraceFilteringMask

Defines the filtering mask for each plugin. The supported values for this registry value change depending on the status of the software component for which you define the registry value. Two values are predefined: 0, all information is filtered (display no information); 0x0fffffffff, no information is filtered (display all information).

Type: REG_DWORD

Default: 0

Note: We recommend that you do not change the value of this registry entry yourself. For assistance, contact CA Support at http://ca.com/support.

TraceFolderPath

Defines the full pathname to the trace file.

Type: REG_SZ

Default: Blank

TraceOutputMask

Defines the filtering mask for the trace output channels - debug stream, file, or ETW. You can specify that the trace outputs to file, to DbgView debug channel, or to WinDbg debug channel. A value of 0 disables any output.

Type: REG_DWORD

Default: 0

Note: We recommend that you do not change the value of this registry entry yourself. For assistance, contact CA Support at http://ca.com/support.

UnloadIfNoPlugins

Specifies whether cainstrm.dll is automatically unloaded (1) when no plug-ins are assigned for a current process.

If you specify 0, cainstrm.dll loads but does not process plug-ins.

Type: REG_DWORD

Default: 1

.NET

CA Access Control maintains .NET settings it uses under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Instrumentation\.NET
```

The Instrumentation\.NET registry key does not contain any registry entries. It contains registry subkeys for the .NET profiler.

Profiler

CA Access Control maintains the profiler settings it uses under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Instrumentation\.NET
\Profiler
```

The Instrumentation\.NEt\Profiler registry key contains the following registry entries:

ApplyOnProcess

Defines the list of processes to which instrumentation applies.

You can define the name of the service or the full pathname. Names are *not* case sensitive. For example, "services.exe", "\system32\services.exe", "c:\windows\system32\services.exe".

Type: REG_MULTI_SZ

Default: w3wp.exe MultiCLRs.exe

CLSID

Defines the CLSID of the profiler.

Type: REG_SZ

Default: {753C5090-0ADD-41B9-B074-8B9A7B833D7E}

OperationMode

Specifies whether to load the profiler into memory.

Type: REG_DWORD

Limits: 0,1

Default: 1

ReadConfigPeriodSec

Specifies the interval to pool the registry for changes.
Type: REG_DWORD

Default: 0x600

TraceDbgEnable

Specifies whether to trace status flag for the cainstrm module, that is, enables tracing into DbgView or Kernel Debugger.

Type: RED_DWORD

Limits: 0, false; 1, true.

Default: 0

TraceFileEnable

Enables tracing into the file

Type: REG_DWORD

Default: 0 (disabled)

TraceFileIsCyclic

Specifies the type of the trace file.

Type: REG_DWORD

Limits: 0, trace file is not cyclic; 1, trace file is cyclic.

Default: 0

TraceFileSizeLimit

Defines the maximum size of the trace file in bytes. A value of 0 means no maximum size limit is imposed on the trace file.

Type: REG_DWORD

Default: 0

TraceFilteringMask

Defines the filtering mask for each plugin. The supported values for this registry value change depending on the status of the software component for which you define the registry value. Two values are predefined: 0, all information is filtered (display no information); 0x0fffffffff, no information is filtered (display all information).

Type: REG_DWORD

Default: 0

Note: We recommend that you do not change the value of this registry entry yourself. For assistance, contact CA Support at http://ca.com/support.

TraceFolderPath

Defines the full pathname to the trace file.

Type: REG_SZ

Default: Blank

TraceOutputMask

Defines the filtering mask for the trace output channels - debug stream, file, or ETW. You can specify that the trace outputs to file, to DbgView debug channel, or to WinDbg debug channel. A value of 0 disables any output.

Type: REG_DWORD

Default: 0

Note: We recommend that you do not change the value of this registry entry yourself. For assistance, contact CA Support at http://ca.com/support.

TraceReadParamsSec

Defines the time interval for updating the trace parameters: WinServicePlg.dll reads updates trace parameters every TraceReadParamsSec.

Type: REG_DWORD

Assemblies

CA Access Control maintains .NET profiler assemblies settings it uses under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Instrumentation\.NET
\Profiler\Assemblies
```

The .NET\Profiler\Assemblies registry key does not contain any registry entries. It contains registry subkeys for the .NET profiler assemblies.

CAPUPM.NETDBPlg

CA Access Control maintains the CAPUPM.NETBDPlg settings it uses under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Instrumentation\.NET
\Profiler\Assemblies\CAPUPM.NETDBPlg
```

The Instrumentation\.NEt\Profiler\Assemblies\CAPUPM.NETDBPlg registry key contains the following registry entries:

BuildNumber

Defines the .NET assembly build version.

Type: REG_DWORD

Default: 0

MajorVersion

Defines the .NET assembly major version number.

Type: REG_DWORD

Default: 1

MinorVersion

Defines the .NET assembly minor version number.

Type: REG_DWORD

Default: 0

PublicKeyToken

Defines the .NET assembly public key token.

Type: REG_BINARY

Default: 5e 84 2e 72 e9 8c 10 e0

RevisionNumber

Defines the .NET assembly revision number.

Type: REG_DWORD
Default: 0

Plugins

CA Access Control maintains .NET profiler plugins settings it uses under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Instrumentation\.NET
\Profiler\Plugins
```

The Instrumentation\.NET\Profiler\Plugins registry key does not contain any registry entries. It contains registry subkeys for the .NET profiler plugins.

DB

CA Access Control maintains the DB settings it uses under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Instrumentation\.NET
\Profiler\Plugins\DB
```

The Instrumentation\.NET\Profiler\Plugins\DB registry key contains the following registry entries:

Altitude

Defines the order of plug-in loading.
Type: REG_DWORD

Default: 1

ApplyOnProcess

Defines the list of processes to which instrumentation applies.

You can define the name of the service or the full pathname. Names are *not* case sensitive. For example, "services.exe", "\system32\services.exe", "c:\windows\system32\services.exe".

Type: REG_MULTI_SZ

Default: w3wp.exe MultiCLRs.exe

AutoBlockNativeAssemblies

Defines whether to block the CAPUPMProfilerDBPlg.dll from loading and load the byte code backup.

Type: REG_DWORD

Default: 1

OperationMode

Specifies whether to load the CAPUPMProfilerDBPlg.dll plugin into memory.

Type: REG_DWORD

Limits: 0,1

Default: 1

PluginPath

Specifies the full pathname of the CAPUPMProfilerDBPlg.dll plugin.

Type: REG_SZ

Default: C:\Program Files\CA\AccessControl\bin\CAPUPMProfilerDBPlg.dll

TraceDbgEnable

Specifies whether to trace status flag for the cainstrm module, that is, enables tracing into DbgView or Kernel Debugger.

Type: RED_DWORD

Limits: 0, false; 1, true.

Default: 0

TraceFileEnable

Enables tracing into the file

Type: REG_DWORD

Default: 0 (disabled)

TraceFileIsCyclic

Specifies the type of the trace file.

Type: REG_DWORD

Limits: 0, trace file is not cyclic; 1, trace file is cyclic.

Default: 0

TraceFileSizeLimit

Defines the maximum size of the trace file in bytes. A value of 0 means no maximum size limit is imposed on the trace file.

Type: REG_DWORD

Default: 0

TraceFilteringMask

Defines the filtering mask for each plugin. The supported values for this registry value change depending on the status of the software component for which you define the registry value. Two values are predefined: 0, all information is filtered (display no information); 0x0fffffffff, no information is filtered (display all information).

Type: REG_DWORD

Default: 0

Note: We recommend that you do not change the value of this registry entry yourself. For assistance, contact CA Support at http://ca.com/support.

TraceFolderPath

Defines the full pathname to the trace file.

Type: REG_SZ

Default: Blank

TraceOutputMask

Defines the filtering mask for the trace output channels - debug stream, file, or ETW. You can specify that the trace outputs to file, to DbgView debug channel, or to WinDbg debug channel. A value of 0 disables any output.

Type: REG_DWORD

Default: 0

Note: We recommend that you do not change the value of this registry entry yourself. For assistance, contact CA Support at http://ca.com/support.

PluginManagement

CA Access Control maintains dynamic plug-in load and unload settings it uses under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\
Instrumentation\PluginManagement
```

The Instrumentation\PluginManagement registry key contains the following registry entries:

Active

Specifies whether plug-ins dynamic loading is active (1).

Type: REG_DWORD.

Default: 1

Altitude

Defines the order of dynamic management stubs in the chain.

Type: REG-DWORD

Default: 0x0fffffff (reserved value)

ApplyOnDLL

Read only value.

Default: Kernel32.dll

ApplyOnProcess

Defines the list of processes to which dynamic loading applies.

You can define the name of the service or the full pathname. Names are *not* case sensitive. For example, "services.exe", "\system32\services.exe", "c:\windows\system32\services.exe".

Type: REG_MULTI_SZ

By default, this token is not set (dynamic loading applies to any plug-in).

ExcludeProcess

Defines the list of processes to which dynamic loading does *not* apply.

Note: This entry is valid only if ApplyOnProcess is not set.

Type: REG_MULTI_SZ

By default, this token is not set.

LoadLibraryA

For internal use only.

Default: 0

LoadLibraryExA

For internal use only.

Default: 0

LoadLibraryExW

For internal use only.

Default: 1

LoadLibraryW

For internal use only

Default: 0

OperationMode

For internal use only.

Default: 1

ProcessCommanArguments

Specifies whether the instrumentation module notifies the CA Access Control security service on process creation event.

Type: REG_DWORD

Values:

0—Instrumentation module does not notify CA Access Control security service on process creation.

1—Instrumentation module notifies CA Access Control security service on process creation.

Note: The registry key value is automatically changed by the CA Access Control security service depending on configuration settings and database definitions. Do not alter the registry key value manually.

PluginName

Read only value.

Default: *ACInstallDir*\bin\cainstrm.dll

PlugIns

CA Access Control maintains plug-in settings it uses under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Instrumentation\PlugIns
```

The Instrumentation\PlugIns registry key does not contain any registry entries. It contains registry subkeys for every loaded plug-in.

CMDPlg

CA Access Control maintains the CMD plug-in settings it uses under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Instrumentation\PlugIns\CMDPlg

The Instrumentation\PlugIns\CMDPlg registry key contains the following registry entries:

Altitude

Defines the order of plug-in loading.

Limits: 1-1000 (values below and above the limits are reserved for internal purposes)

Type: REG_DWORD

Default: 5

ApplyOnDLL

Defines the DLL names (modules) to which the current plug-in applies.

Type: REG_SZ

Default: Kernel32.dll

ApplyOnProcess

Defines the processes to which the current plug-in applies.

You can provide the name of the service, the filename, or the full pathname. For example, "services.exe", "\system32\services.exe", "c:\windows\system32\services.exe".

Type: REG_MULTI_SZ

Note: If this registry entry has only one value, REG_SZ is also a valid type.

Default: CMD.exe

CommunicationWaitTimeout

Defines the maximum time, in seconds, that the plug-in waits when it sends or receives transactions.

Type: REG_DWORD

Default: 15

ExcludeProcess

Defines the processes to which the plug-in does not apply.

Note: This entry is valid only if ApplyOnProcess is not set.

Type: REG_MULTI_SZ

By default, this is empty.

OperationMode

Specifies whether to load the plug-in (1) into memory.

Type: REG_DWORD

Default: 1

PluginName

Defines the name of the dynamic link library (DLL) for the plug-in.

Type: REG_SZ

Default: *ACInstallDir*\bin\CMDPlg.dll

ServiceTimeOut

Defines in milliseconds the maximum interval to wait for a transaction with seosd.

Note: If the timeout expires, the request is authorized.

Type: REG_DWORD

Default: 0x00000bb8 (3000 decimal)

TraceDbgEnable

Specifies whether to trace status flag for the cainstrm module, that is, enables tracing into DbgView or Kernel Debugger.

Type: RED_DWORD

Limits: 0, false; 1, true.

Default: 0

TraceFileEnable

Enables tracing into the file

Type: REG_DWORD

Default: 0 (disabled)

TraceFileIsCyclic

Specifies the type of the trace file.

Type: REG_DWORD

Limits: 0, trace file is not cyclic; 1, trace file is cyclic.

Default: 0

TraceFileSizeLimit

Defines the maximum size of the trace file in bytes. A value of 0 means no maximum size limit is imposed on the trace file.

Type: REG_DWORD

Default: 0

TraceFilteringMask

Defines the filtering mask for each plugin. The supported values for this registry value change depending on the status of the software component for which you define the registry value. Two values are predefined: 0, all information is filtered (display no information); 0x0fffffffff, no information is filtered (display all information).

Type: REG_DWORD

Default: 0

Note: We recommend that you do not change the value of this registry entry yourself. For assistance, contact CA Support at http://ca.com/support.

TraceFolderPath

Defines the full pathname to the trace file.

Type: REG_SZ

Default: Blank

TraceOutputMask

Defines the filtering mask for the trace output channels - debug stream, file, or ETW. You can specify that the trace outputs to file, to DbgView debug channel, or to WinDbg debug channel. A value of 0 disables any output.

Type: REG_DWORD

Default: 0

Note: We recommend that you do not change the value of this registry entry yourself. For assistance, contact CA Support at http://ca.com/support.

TraceReadParamsSec

Defines the time interval for updating the trace parameters: WinServicePlg.dll reads updates trace parameters every TraceReadParamsSec.

Type: REG_DWORD

Default: 60

OCIPlg

CA Access Control maintains the OCI plug-in settings it uses under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Instrumentation\PlugIns\OCIPlg

The Instrumentation\PlugIns\OCIPlg registry key contains the following registry entries:

Altitude

Defines the order of plug-in loading.

Limits: 1-1000 (values below and above the limits are reserved for internal purposes)

Type: REG_DWORD

Default: 5

ApplyOnDLL

Defines the DLL names (modules) to which the current plug-in applies.

Type: REG_SZ

Default: oci.dll

ApplyOnProcess

Defines the processes to which the current plug-in applies.

You can provide the name of the service, the filename, or the full pathname. For example, "services.exe", "\system32\services.exe", "c:\windows\system32\services.exe".

Type: REG_MULTI_SZ

Note: If this registry entry has only one value, REG_SZ is also a valid type.

Default: sqlplus.exe w3wp.exe

CommunicationWaitTimeout

Defines the maximum time, in seconds, that the plug-in waits when it sends or receives transactions.

Type: REG_DWORD

Default: 15

EnvironmentVariables

Specifies the environment variables that are forwarded to the Privileged User Password Management Agent

Type: REG_MULTI_SZ

Default: TNS_ADMIN ORACLE_HOME

Note: We recommend that you do not change the value of this registry entry yourself. For assistance, contact CA Support at http://ca.com/support.

ExcludeProcess

Defines the processes to which the plug-in does not apply.

Note: This entry is valid only if ApplyOnProcess is not set.

Type: REG_MULTI_SZ

By default, this is empty.

OperationMode

Specifies whether to load the plug-in (1) into memory.

Type: REG_DWORD

Default: 0

PluginName

Defines the name of the dynamic link library (DLL) for the plug-in.

Type: REG_SZ

Default: *ACInstallDir*\bin\OCIPlg.dll

TraceDbgEnable

Specifies whether to trace status flag for the cainstrm module, that is, enables tracing into DbgView or Kernel Debugger.

Type: RED_DWORD

Limits: 0, false; 1, true.

Default: 0

TraceFileEnable

Enables tracing into the file

Type: REG_DWORD

Default: 0 (disabled)

TraceFileIsCyclic

Specifies the type of the trace file.

Type: REG_DWORD

Limits: 0, trace file is not cyclic; 1, trace file is cyclic.

Default: 0

TraceFileSizeLimit

Defines the maximum size of the trace file in bytes. A value of 0 means no maximum size limit is imposed on the trace file.

Type: REG_DWORD

Default: 0

TraceFilteringMask

Defines the filtering mask for each plugin. The supported values for this registry value change depending on the status of the software component for which you define the registry value. Two values are predefined: 0, all information is filtered (display no information); 0x0fffffffff, no information is filtered (display all information).

Type: REG_DWORD

Default: 0

Note: We recommend that you do not change the value of this registry entry yourself. For assistance, contact CA Support at http://ca.com/support.

TraceFolderPath

Defines the full pathname to the trace file.

Type: REG_SZ

Default: Blank

TraceOutputMask

Defines the filtering mask for the trace output channels - debug stream, file, or ETW. You can specify that the trace outputs to file, to DbgView debug channel, or to WinDbg debug channel. A value of 0 disables any output.

Type: REG_DWORD

Default: 0

Note: We recommend that you do not change the value of this registry entry yourself. For assistance, contact CA Support at http://ca.com/support.

TraceReadParamsSec

Defines the time interval for updating the trace parameters: WinServicePlg.dll reads updates trace parameters every TraceReadParamsSec.

Type: REG_DWORD

Default: 60

UpgradeWaitTimeOutMaxTries

Specifies the number of retry attempts to update the plugin.

Type: REG_DWORD

Default: 3

Note: We recommend that you do not change the value of this registry entry yourself. For assistance, contact CA Support at http://ca.com/support.

UpgradeWaitTimeOutMilliseconds

Specifies the timeout period, in milliseconds, to declare a failure to upgrade.

Type: REG_DWORD

Default: 0x1ffff (131071)

Note: We recommend that you do not change the value of this registry entry yourself. For assistance, contact CA Support at http://ca.com/support.

ODBCPlg

CA Access Control maintains the Privileged User Password Management Agent ODBC plug-in settings it uses under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Instrumentation\PlugIns\ODBCPlg
```

The Instrumentation\PlugIns\ODBCPlg registry key contains the following registry entries:

Altitude

Defines the order of plug-in loading.

Limits: 1-1000 (values below and above the limits are reserved for internal purposes)

Type: REG_DWORD

Default: 5

ApplyOnDLL

Defines the DLL names (modules) to which the current plug-in applies.

Type: REG_MULTI_SZ

Default: ODBC32.dll

ApplyOnProcess

Defines the processes to which the current plug-in applies.

You can provide the name of the service, the filename, or the full pathname. For example, "services.exe", "\system32\services.exe", "c:\windows\system32\services.exe".

Type: REG_MULTI_SZ

Note: If this registry entry has only one value, REG_SZ is also a valid type.

Default: w3wp.exe

CommunicationWaitTimeout

Defines the maximum time, in seconds, that the plug-in waits when it sends or receives transactions.

Type: REG_DWORD

Default: 15

EnvironmentVariables

Specifies the environment variables that are forwarded to the Privileged User Password Management Agent

Type: REG_MULTI_SZ

Default: TNS_ADMIN ORACLE_HOME

Note: We recommend that you do not change the value of this registry entry yourself. For assistance, contact CA Support at http://ca.com/support.

ExcludeProcess

Defines the processes to which the plug-in does not apply.

Note: This entry is valid only if ApplyOnProcess is not set.

Type: REG_MULTI_SZ

By default, this is empty.

OperationMode

Specifies whether to load the plug-in (1) into memory.

Type: REG_DWORD

Default: 0

PluginName

Defines the name of the dynamic link library (DLL) for the plug-in.

Type: REG_SZ

Default: *ACInstallDir*\bin\ODBCPlg.dll

TraceDbgEnable

Specifies whether to trace status flag for the cainstrm module, that is, enables tracing into DbgView or Kernel Debugger.

Type: RED_DWORD

Limits: 0, false; 1, true.

Default: 0

TraceFileEnable

Enables tracing into the file

Type: REG_DWORD

Default: 0 (disabled)

TraceFileIsCyclic

Specifies the type of the trace file.

Type: REG_DWORD

Limits: 0, trace file is not cyclic; 1, trace file is cyclic.

Default: 0

TraceFileSizeLimit

Defines the maximum size of the trace file in bytes. A value of 0 means no maximum size limit is imposed on the trace file.

Type: REG_DWORD

Default: 0

TraceFilteringMask

Defines the filtering mask for each plugin. The supported values for this registry value change depending on the status of the software component for which you define the registry value. Two values are predefined: 0, all information is filtered (display no information); 0x0fffffffff, no information is filtered (display all information).

Type: REG_DWORD

Default: 0

Note: We recommend that you do not change the value of this registry entry yourself. For assistance, contact CA Support at http://ca.com/support.

TraceFolderPath

Defines the full pathname to the trace file.

Type: REG_SZ

Default: Blank

TraceOutputMask

Defines the filtering mask for the trace output channels - debug stream, file, or ETW. You can specify that the trace outputs to file, to DbgView debug channel, or to WinDbg debug channel. A value of 0 disables any output.

Type: REG_DWORD

Default: 0

Note: We recommend that you do not change the value of this registry entry yourself. For assistance, contact CA Support at http://ca.com/support.

TraceReadParamsSec

Defines the time interval for updating the trace parameters: WinServicePlg.dll reads updates trace parameters every TraceReadParamsSec.

Type: REG_DWORD

Default: 60

UpgradeWaitTimeOutMaxTries

Specifies the number of retry attempts to update the plugin.

Type: REG_DWORD

Default: 3

Note: We recommend that you do not change the value of this registry entry yourself. For assistance, contact CA Support at http://ca.com/support.

UpgradeWaitTimeOutMilliseconds

Specifies the timeout period, in milliseconds, to declare a failure to upgrade.

Type: REG_DWORD

Default: 0x1ffff (131071)

Note: We recommend that you do not change the value of this registry entry yourself. For assistance, contact CA Support at http://ca.com/support.

OLEDBPlg

CA Access Control maintains the Privileged User Password Management Agent OLEDB plug-in settings it uses under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Instrumentation\PlugIns\OLEDBlg
```

The Instrumentation\PlugIns\OLEDBPlg registry key contains the following registry entries:

Altitude

Defines the order of plug-in loading.

Limits: 1-1000 (values below and above the limits are reserved for internal purposes)

Type: REG_DWORD

Default: 5

ApplyOnDLL

Defines the DLL names (modules) to which the current plug-in applies.

Type: REG_MULTI_SZ

Default: kernel32.dll

ApplyOnProcess

Defines the processes to which the current plug-in applies.

You can provide the name of the service, the filename, or the full pathname. For example, "services.exe", "\system32\services.exe", "c:\windows\system32\services.exe".

Type: REG_MULTI_SZ

Note: If this registry entry has only one value, REG_SZ is also a valid type.

Default: w3wp.exe sqlcmd.exe

CommunicationWaitTimeout

Defines the maximum time, in seconds, that the plug-in waits when it sends or receives transactions.

Type: REG_DWORD

Default: 15

EnvironmentVariables

Specifies the environment variables that are forwarded to the Privileged User Password Management Agent

Type: REG_MULTI_SZ

Default: TNS_ADMIN ORACLE_HOME

Note: We recommend that you do not change the value of this registry entry yourself. For assistance, contact CA Support at http://ca.com/support.

ExcludeProcess

Defines the processes to which the plug-in does not apply.

Note: This entry is valid only if ApplyOnProcess is not set.

Type: REG_MULTI_SZ

By default, this is empty.

OperationMode

Specifies whether to load the plug-in (1) into memory.

Type: REG_DWORD

Default: 0

PluginName

Defines the name of the dynamic link library (DLL) for the plug-in.

Type: REG_SZ

Default: *ACInstallDir*\bin\OLEDBPlg.dll

SerializationWaitTimeout

Defines internal synchronization of loadlibrary and DllGetClassObject class.

Type: REG_DWORD

Default: 0xa (10 decimal)

Note: We recommend that you do not change the value of this registry entry yourself. For assistance, contact CA Support at http://ca.com/support.

TraceDbgEnable

Specifies whether to trace status flag for the cainstrm module, that is, enables tracing into DbgView or Kernel Debugger.

Type: RED_DWORD

Limits: 0, false; 1, true.

Default: 0

TraceFileEnable

Enables tracing into the file

Type: REG_DWORD

Default: 0 (disabled)

TraceFileIsCyclic

Specifies the type of the trace file.

Type: REG_DWORD

Limits: 0, trace file is not cyclic; 1, trace file is cyclic.

Default: 0

TraceFileSizeLimit

Defines the maximum size of the trace file in bytes. A value of 0 means no maximum size limit is imposed on the trace file.

Type: REG_DWORD

Default: 0

TraceFilteringMask

Defines the filtering mask for each plugin. The supported values for this registry value change depending on the status of the software component for which you define the registry value. Two values are predefined: 0, all information is filtered (display no information); 0x0fffffffff, no information is filtered (display all information).

Type: REG_DWORD

Default: 0

Note: We recommend that you do not change the value of this registry entry yourself. For assistance, contact CA Support at http://ca.com/support.

TraceFolderPath

Defines the full pathname to the trace file.

Type: REG_SZ

Default: Blank

TraceOutputMask

Defines the filtering mask for the trace output channels - debug stream, file, or ETW. You can specify that the trace outputs to file, to DbgView debug channel, or to WinDbg debug channel. A value of 0 disables any output.

Type: REG_DWORD

Default: 0

Note: We recommend that you do not change the value of this registry entry yourself. For assistance, contact CA Support at http://ca.com/support.

TraceReadParamsSec

Defines the time interval for updating the trace parameters: WinServicePlg.dll reads updates trace parameters every TraceReadParamsSec.

Type: REG_DWORD

Default: 60

UpgradeWaitTimeOutMaxTries

Specifies the number of retry attempts to update the plugin.

Type: REG_DWORD

Default: 3

Note: We recommend that you do not change the value of this registry entry yourself. For assistance, contact CA Support at http://ca.com/support.

UpgradeWaitTimeOutMilliseconds

Specifies the timeout period, in milliseconds, to declare a failure to upgrade.

Type: REG_DWORD

Default: 0x1ffff (131071)

Note: We recommend that you do not change the value of this registry entry yourself. For assistance, contact CA Support at http://ca.com/support.

Providers

CA Access Control maintains settings for providers that the OLEDB plug-in supports under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Instrumentation\Plug Ins\
OLEDBPlg\Providers
```

The Instrumentation\PlugIns\OLEDBPlg\Providers registry key does not contain any registry entries. The key contains registry subkeys for every provider that the OLEDB plug-in supports.

Note: Some providers that the OLEDB plug-in supports are not supported in CA Access Control.

Generic

CA Access Control maintains settings for generic providers that the OLEDB plug-in supports under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Instrumentation\Plug Ins\
OLEDBPlg\Providers\Generic
```

The Instrumentation\PlugIns\OLEDBPlg\Providers\Generic registry key does not contain any registry entries. The key contains registry subkeys for generic providers that the OLEDB plug-in supports.

CLSID

CA Access Control maintains CLSID (class identifier) settings for generic providers that the OLEDB plug-in supports under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Instrumentation\Plug Ins\
OLEDBPlg\Providers\Generic\CLSID
```

By default, the Instrumentation\PlugIns\OLEDBPlg\Providers\Generic\CLSID registry key does not contain any registry entries. Entries that you create in this subkey must have the following format:

CLSID

Defines the class identifier for the provider.

Type: REG_SZ

Limits: 1, enable support for the provider; 0, disable support for the provider.

Name

CA Access Control maintains settings for generic providers that the OLEDB plug-in supports under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Instrumentation\PlugIns\
OLEDBPlg\Providers\Generic\Name
```

The Instrumentation\PlugIns\OLEDBPlg\Providers\Generic\Name registry key contains the following registry entries:

Microsoft OLE DB Provider for ODBC Drivers

Specifies that the OLEDB plug-in supports the Microsoft OLE DB Provider for ODBC Drivers.

Type: REG_DWORD

Limits: 1, enable support; 0, disable support.

Default: 1

Jet

CA Access Control maintains settings for Microsoft Jet-based providers that the OLEDB plug-in supports under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Instrumentation\PlugIns\
OLEDBPlg\Providers\Jet
```

The Instrumentation\PlugIns\OLEDBPlg\Providers\Jet registry key does not contain any registry entries. The key contains registry subkeys for Microsoft Jet-based providers that the OLEDB plug-in supports.

Note: CA Access Control currently does not support Microsoft Jet-based providers.

CLSID

CA Access Control maintains CLSID (class identifier) settings for Microsoft Jet-based providers that the OLEDB plug-in supports under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Instrumentation\Plug Ins\
OLEDBPlg\Providers\Jet\CLSID
```

Note: CA Access Control currently does not support Microsoft Jet-based providers.

By default, the Instrumentation\PlugIns\OLEDBPlg\Providers\Jet\CLSID registry key does not contain any registry entries. Entries that you create in this subkey must have the following format:

CLSID

Defines the class identifier for the provider.

Type: REG_SZ

Limits: 1, enable support for the provider; 0, disable support for the provider.

Name

CA Access Control maintains settings for Microsoft Jet-based providers that the OLEDB plug-in supports under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Instrumentation\Plug Ins\
OLEDBPlg\Providers\Jet\Name
```

Note: CA Access Control currently does not support Microsoft Jet-based providers.

The Instrumentation\PlugIns\OLEDBPlg\Providers\Jet\Name registry key contains the following registry entries:

Microsoft Jet 4.0 OLE DB Provider

Specifies that the OLEDB plug-in supports the Microsoft Jet 4.0 OLE DB Provider.

Type: REG_DWORD

Limits: 1, enable support; 0, disable support.

Default: 1

Microsoft Office 12.0 Access Database Engine OLE DB Provider

Specifies that the OLEDB plug-in supports the Microsoft Office 12.0 Access Database Engine OLE DB Provider.

Type: REG_DWORD

Limits: 1, enable support; 0, disable support.

Default: 1

MSSQL

CA Access Control maintains settings for Microsoft SQL Server-based providers that the OLEDB plug-in supports under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Instrumentation\Plug Ins\
OLEDBPlg\Providers\MSSQL
```

The Instrumentation\PlugIns\OLEDBPlg\Providers\MSSQL registry key does not contain any registry entries. The key contains registry subkeys for Microsoft SQL Server-based providers that the OLEDB plug-in supports.

CLSID

CA Access Control maintains CLSID (class identifier) settings for Microsoft SQL Server-based providers that the OLEDB plug-in supports under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Instrumentation\Plug Ins\
OLEDBPlg\Providers\MSSQL\CLSID
```

By default, the Instrumentation\PlugIns\OLEDBPlg\Providers\MSSQL\CLSID registry key does not contain any registry entries. Entries that you create in this subkey must have the following format:

CLSID

Defines the class identifier for the provider.

Type: REG_SZ

Limits: 1, enable support for the provider; 0, disable support for the provider.

Name

CA Access Control maintains settings for Microsoft SQL Server-based providers that the OLEDB plug-in supports under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Instrumentation\PlugIns\
OLEDBPlg\Providers\MSSQL\Name
```

The Instrumentation\PlugIns\OLEDBPlg\Providers\MSSQL\Name registry key contains the following registry entries:

Microsoft OLE DB Provider for SQL Server

Specifies that the OLEDB plug-in supports the Microsoft OLE DB Provider for SQL Server.

Type: REG_DWORD

Limits: 1, enable support; 0, disable support.

Default: 1

SQL Native Client

Specifies that the OLEDB plug-in supports the SQL Native Client provider.

Type: REG_DWORD

Limits: 1, enable support; 0, disable support.

Default: 1

SQL Server Native Client 10.0

Specifies that the OLEDB plug-in supports the SQL Server Native Client 10.0 provider.

Type: REG_DWORD

Limits: 1, enable support; 0, disable support.

Default: 1

MySQL

CA Access Control maintains settings for MySQL-based providers that the OLEDB plug-in supports under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Instrumentation\PlugIns\
OLEDBPlg\Providers\MySQL
```

The Instrumentation\PlugIns\OLEDBPlg\Providers\MySQL registry key does not contain any registry entries. The key contains registry subkeys for MySQL-based providers that the OLEDB plug-in supports.

Note: CA Access Control currently does not support MySQL-based providers.

CLSID

CA Access Control maintains CLSID (class identifier) settings for MySQL-based providers that the OLEDB plug-in supports under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Instrumentation\PlugIns\
OLEDBPlg\Providers\MySQL\CLSID
```

Note: CA Access Control currently does not support MySQL-based providers.

By default, the Instrumentation\PlugIns\OLEDBPlg\Providers\MySQL\CLSID registry key does not contain any registry entries. Entries that you create in this subkey must have the following format:

CLSID

Defines the class identifier for the provider.

Type: REG_SZ

Limits: 1, enable support for the provider; 0, disable support for the provider.

Name

CA Access Control maintains settings for MySQL-based providers that the OLEDB plug-in supports under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Instrumentation\Plug Ins\
OLEDBPlg\Providers\MySQL\Name
```

Note: CA Access Control currently does not support MySQL-based providers.

The Instrumentation\PlugIns\OLEDBPlg\Providers\MySQL\Name registry key contains the following registry entries:

MySQL Provider

Specifies that the OLEDB plug-in supports the MySQL Provider.

Type: REG_DWORD

Limits: 1, enable support; 0, disable support.

Default: 1

MySQL.OLEDB Provider

Specifies that the OLEDB plug-in supports the MySQL.OLEDB Provider.

Type: REG_DWORD

Limits: 1, enable support; 0, disable support.

Default: 1

Oracle

CA Access Control maintains settings for Oracle-based providers that the OLEDB plug-in supports under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Instrumentation\Plug Ins\
OLEDBPlg\Providers\Oracle
```

The Instrumentation\PlugIns\OLEDBPlg\Providers\Oracle registry key does not contain any registry entries. The key contains registry subkeys for Oracle-based providers that the OLEDB plug-in supports.

CLSID

CA Access Control maintains CLSID (class identifier) settings for Oracle-based providers that the OLEDB plug-in supports under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Instrumentation\PlugIns\
OLEDBPlg\Providers\Oracle\CLSID
```

By default, the Instrumentation\PlugIns\OLEDBPlg\Providers\Oracle\CLSID registry key does not contain any registry entries. Entries that you create in this subkey must have the following format:

CLSID

Defines the class identifier for the provider.

Type: REG_SZ

Limits: 1, enable support for the provider; 0, disable support for the provider.

Name

CA Access Control maintains settings for Oracle-based providers that the OLEDB plug-in supports under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Instrumentation\PlugIns\
OLEDBPlg\Providers\Oracle\Name
```

The Instrumentation\PlugIns\OLEDBPlg\Providers\Oracle\Name registry key contains the following registry entries:

Microsoft OLE DB Provider for Oracle

Specifies that the OLEDB plug-in supports the Microsoft OLE DB Provider for Oracle.

Type: REG_DWORD

Limits: 1, enable support; 0, disable support.

Default: 1

Oracle Provider for OLE DB

Specifies that the OLEDB plug-in supports the Oracle Provider for OLE DB.

Type: REG_DWORD

Limits: 1, enable support; 0, disable support.

Default: 1

RunAsPlg

CA Access Control maintains the RunAs plug-in settings it uses under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Instrumentation\PlugIns\RunAsPlg

The Instrumentation\PlugIns\RunAsPlg registry key contains the following registry entries:

Altitude

Defines the order of plug-in loading.

Limits: 1-1000 (values below and above the limits are reserved for internal purposes)

Type: REG_DWORD

Default: 5

ApplyOnDLL

Defines the DLL names (modules) to which the current plug-in applies.

Type: REG_MULTI_SZ

Default: advapi32.dll

ApplyOnProcess

Defines the processes to which the current plug-in applies.

You can provide the name of the service, the filename, or the full pathname. For example, "services.exe", "\system32\services.exe", "c:\windows\system32\services.exe".

Type: REG_MULTI_SZ

Note: If this registry entry has only one value, REG_SZ is also a valid type.

Default: runas.exe explorer.exe consent.exe

Note: The consent.exe value applies to only Windows Server 2008 computers.

CommunicationWaitTimeout

Defines the maximum time, in seconds, that the plug-in waits when it sends or receives transactions.

Type: REG_DWORD

Default: 15

ExcludeProcess

Defines the processes to which the plug-in does not apply.

Note: This entry is valid only if ApplyOnProcess is not set.

Type: REG_MULTI_SZ

By default, this is empty.

OperationMode

Specifies whether to load the plug-in (1) into memory.

Type: REG_DWORD

Default: 1

PluginName

Defines the name of the dynamic link library (DLL) for the plug-in.

Type: REG_SZ

Default: *ACInstallDir*\bin\RunAsPlg.dll

ServiceTimeOut

Defines in milliseconds the maximum interval to wait for a transaction with seosd.

Note: If the timeout expires, the request is authorized.

Type: REG_DWORD

Default: 0x00000bb8 (3000 decimal)

TraceDbgEnable

Specifies whether to trace status flag for the cainstrm module, that is, enables tracing into DbgView or Kernel Debugger.

Type: RED_DWORD

Limits: 0, false; 1, true.

Default: 0

TraceFileEnable

Enables tracing into the file

Type: REG_DWORD

Default: 0 (disabled)

TraceFileIsCyclic

Specifies the type of the trace file.

Type: REG_DWORD

Limits: 0, trace file is not cyclic; 1, trace file is cyclic.

Default: 0

TraceFileSizeLimit

Defines the maximum size of the trace file in bytes. A value of 0 means no maximum size limit is imposed on the trace file.

Type: REG_DWORD

Default: 0

TraceFilteringMask

Defines the filtering mask for each plugin. The supported values for this registry value change depending on the status of the software component for which you define the registry value. Two values are predefined: 0, all information is filtered (display no information); 0x0fffffffff, no information is filtered (display all information).

Type: REG_DWORD

Default: 0

Note: We recommend that you do not change the value of this registry entry yourself. For assistance, contact CA Support at http://ca.com/support.

TraceFolderPath

Defines the full pathname to the trace file.

Type: REG_SZ

Default: Blank

TraceOutputMask

Defines the filtering mask for the trace output channels - debug stream, file, or ETW. You can specify that the trace outputs to file, to DbgView debug channel, or to WinDbg debug channel. A value of 0 disables any output.

Type: REG_DWORD

Default: 0

Note: We recommend that you do not change the value of this registry entry yourself. For assistance, contact CA Support at http://ca.com/support.

TraceReadParamsSec

Defines the time interval for updating the trace parameters: WinServicePlg.dll reads updates trace parameters every TraceReadParamsSec.

Type: REG_DWORD

Default: 60

StopPlg

CA Access Control maintains Stack Overflow Protection (STOP) plug-in settings it uses under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Instrumentation\PlugIns\StopPlg
```

The Instrumentation\PlugIns\StopPlg registry key contains the following registry entries:

Altitude

Defines the order of plug-in loading.

Limits: 1-1000 (values below and above the limits are reserved for internal purposes)

Type: REG_DWORD

Default: 5

ApplyOnDLL

Defines the DLL names (modules) to which the current plug-in applies.

Type: REG_MULTI_SZ

Default: Kernel32.dll

ApplyOnProcess

Defines the processes to which the current plug-in applies.

You can provide the name of the service, the filename, or the full pathname. For example, "services.exe", "\system32\services.exe", "c:\windows\system32\services.exe".

Type: REG_MULTI_SZ

Note: If this registry entry has only one value, REG_SZ is also a valid type.

By default, this token is not set (plug-in applies to any process).

ExcludeProcess

Defines the processes to which the plug-in does not apply.

Note: This entry is valid only if ApplyOnProcess is not set.

Type: REG_MULTI_SZ

Default (Windows 2008): slsvc.exe

Default (all other Windows versions): Blank (token is not set)

OperationMode

Specifies whether to load the plug-in (1) into memory.

Type: REG_DWORD

Default: 0

PluginName

Defines the name of the dynamic link library (DLL) for the plug-in.

Type: REG_SZ

Default: *ACInstallDir*\bin\StopPlg.dll

STOPClientTraceEnabled

Specifies whether the STOP client module has trace logging enabled.

Type: REG_DWORD

Default: 0 (disabled)

STOPClientTraceModulePath

Defines the full pathname of the STOP client module trace logging module.

Type: REG_SZ

Default: *ACInstallDir*\bin\STOPClientTrace.dll

STOPSEHHandlingModeDisabled

Specifies whether STOP extensive checks for SEH based exploits are enabled.

Type: REG_DWORD

Default: 1 (disabled)

TraceDbgEnable

Specifies whether to trace status flag for the cainstrm module, that is, enables tracing into DbgView or Kernel Debugger.

Type: RED_DWORD

Limits: 0, false; 1, true.

Default: 0

TraceFileIsCyclic

Specifies the type of the trace file.

Type: REG_DWORD

Limits: 0, trace file is not cyclic; 1, trace file is cyclic.

Default: 0

TraceFileSizeLimit

Defines the maximum size of the trace file in bytes. A value of 0 means no maximum size limit is imposed on the trace file.

Type: REG_DWORD

Default: 0

TraceFilteringMask

Defines the filtering mask for each plugin. The supported values for this registry value change depending on the status of the software component for which you define the registry value. Two values are predefined: 0, all information is filtered (display no information); 0x0fffffffff, no information is filtered (display all information).

Type: REG_DWORD

Default: 0

Note: We recommend that you do not change the value of this registry entry yourself. For assistance, contact CA Support at http://ca.com/support.

TraceFolderPath

Defines the full pathname to the trace file.

Type: REG_SZ

Default: Blank

TraceOutputMask

Defines the filtering mask for the trace output channels - debug stream, file, or ETW. You can specify that the trace outputs to file, to DbgView debug channel, or to WinDbg debug channel. A value of 0 disables any output.

Type: REG_DWORD

Default: 0

Note: We recommend that you do not change the value of this registry entry yourself. For assistance, contact CA Support at http://ca.com/support.

WinServicePlg

CA Access Control maintains Windows services protection plug-in settings it uses under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Instrumentation\
PlugIns\WinServicePlg
```

The Instrumentation\PlugIns\WinServicePlg registry key contains the following registry entries:

Altitude

Defines the order of plug-in loading.

Limits: 1-1000 (values below and above the limits are reserved for internal purposes)

Type: REG_DWORD

Default: 5

ApplyOnDLL

Defines the DLL names (modules) to which the current plug-in applies.

Type: REG_MULTI_SZ

Default: Rpcrt4.dll

ApplyOnProcess

Defines the processes to which the current plug-in applies.

You can provide the name of the service, the filename, or the full pathname. For example, "services.exe", "\system32\services.exe", "c:\windows\system32\services.exe".

Type: REG_MULTI_SZ

Note: If this registry entry has only one value, REG_SZ is also a valid type.

Default: Services.exe

ExcludeProcess

Defines the processes to which the plug-in does not apply.

Note: This entry is valid only if ApplyOnProcess is not set.

Type: REG_MULTI_SZ

By default, this is empty.

OperationMode

Specifies whether to load the plug-in (1) into memory.

Type: REG_DWORD

Default: 1

PluginName

Defines the name of the dynamic link library (DLL) for the plug-in.

Type: REG_SZ

Default: *ACInstallDir*\bin\WinServicePlg.dll

ServiceTimeOut

Defines in milliseconds the maximum interval to wait for a transaction with seosd.

Note: If the timeout expires, the request is authorized.

Type: REG_DWORD

Default: 0x00000bb8 (3000 decimal)

TraceDbgEnable

Specifies whether to trace status flag for the cainstrm module, that is, enables tracing into DbgView or Kernel Debugger.

Type: RED_DWORD

Limits: 0, false; 1, true.

Default: 0

TraceFileEnable

Enables tracing into the file

Type: REG_DWORD

Default: 0 (disabled)

TraceFileIsCyclic

Specifies the type of the trace file.

Type: REG_DWORD

Limits: 0, trace file is not cyclic; 1, trace file is cyclic.

Default: 0

TraceFileSizeLimit

Defines the maximum size of the trace file in bytes. A value of 0 means no maximum size limit is imposed on the trace file.

Type: REG_DWORD

Default: 0

TraceFilteringMask

Defines the filtering mask for each plugin. The supported values for this registry value change depending on the status of the software component for which you define the registry value. Two values are predefined: 0, all information is filtered (display no information); 0x0fffffffff, no information is filtered (display all information).

Type: REG_DWORD

Default: 0

Note: We recommend that you do not change the value of this registry entry yourself. For assistance, contact CA Support at http://ca.com/support.

TraceFolderPath

Defines the full pathname to the trace file.

Type: REG_SZ

Default: Blank

TraceOutputMask

Defines the filtering mask for the trace output channels - debug stream, file, or ETW. You can specify that the trace outputs to file, to DbgView debug channel, or to WinDbg debug channel. A value of 0 disables any output.

Type: REG_DWORD

Default: 0

Note: We recommend that you do not change the value of this registry entry yourself. For assistance, contact CA Support at http://ca.com/support.

TraceReadParamsSec

Defines the time interval for updating the trace parameters: WinServicePlg.dll reads updates trace parameters every TraceReadParamsSec.

Type: REG_DWORD

Default: 0x0000003c (60 decimal)

lang

CA Access Control maintains management language (selang) settings it uses under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\lang
```

The lang registry key contains the following registry entries:

HandleHomeDir

The value that determines whether property HOME_DIR for native user account is updated and home directory created.

If the value is set to 0, only user's property HOME_DIR is updated. If the value is set to 1, user's property is updated and home directory is physically created in the file system.

Default: 1

help_path

The directory in which the lang help files are located.

Default: *ACInstallDir*\data\help

ModifiableClassFlags

Specifies the flags that an CA Access Control administrator can change using the following selang command: setoptions class *className* flags{+ | -} (*flag*)

Values: W—Set Warning mode for the specified class; I—Change case sensitivity for resources in the specified class; WI—Set Warning mode and change case sensitivity for resources in the specified class

Default: W

query_size

The maximum number of records to be listed in a database query.

Default: 100

SetBlockRun

Specifies whether to check if a program is trusted and block the execution of untrusted programs.

Valid values are:

yes-All programs defined with viapgm authorization rules have the blockrun property set to yes.

no-All programs defined with viapgm authorization rules have the blockrun property set to no.

Default: Yes

SpaceReplace

For internal use only. This key should always be empty.

Default: ""

use_old_commands

Specifies whether to disable old ACF2™ compatibility commands (ag, lg, rg, lu, au, and so on).

Limits: 0—do not support old commands, 1—support old commands

Default: 1 (support old commands)

logmgr Key—Registry Settings

CA Access Control maintains logging settings it uses under the following key:

`HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\logmgr`

The logmgr registry key contains the following registry entries:

audit_back

The name of the CA Access Control audit backup file. Only CA Access Control can write to this file.

Default: *ACInstallDir*\log\seos.audit.bak

audit_group

The group that can read the audit logs.

Default: ComputerAssociates

audit_log

The name of the CA Access Control audit log file. When this file reaches the size specified in audit_size, CA Access Control closes the file, renames it with the name in audit_back, and creates a new audit log. Only CA Access Control can write to this file.

Default: *ACInstallDir*\log\seos.audit

audit_max_files

Defines the maximal number of audit log backup files CA Access Control accumulates when it performs date-triggered backups. When the BackUp_Date configuration setting is set to anything other than *none*, CA Access Control continuously accumulates date-triggered backup files. This configuration setting lets you reduce disk space CA Access Control uses for audit log backups. When the number of audit log backup files reaches the limit you set, CA Access Control deletes the oldest backup file when it creates the newest.

Values:

- **0**—keep all audit log backup files.
- *n*—a positive integer greater than zero.

Note: You cannot remove redundant audit log backup files manually because CA Access Control protects these automatically. Also, if the audit reporting is enabled, CA Access Control does not delete a backup file until the Report Agent finishes processing it.

Default: 50

audit_size

The maximum size, in KB, of the CA Access Control audit log file. Do not specify less than 50 KB.

Default: 10240

Note: CA Access Control stops writing audit records to the audit file when the audit file size exceeds 2 GB.

AuditFiltersFile

The name of the CA Access Control audit filter file.

Default: *ACInstallDir*\data\audit.cfg

BackUp_Date

Specifies the criterion by which CA Access Control backs up the audit log file, and if CA Access Control adds a timestamp to the backup file name.

CA Access Control *always* backs up the audit log file when it reaches the size specified in the audit_size configuration setting.

Values: none, yes, daily, weekly, monthly

- yes—CA Access Control backs up the audit log file when it reaches the size specified in audit_size and adds a timestamp to the backup file name.
- none—CA Access Control backs up the audit log file when it reaches the size specified in audit_size and does not add a timestamp to the backup file name.

- daily, weekly, monthly——CA Access Control backs up the audit log file whenever the specified interval has elapsed *and* when it reaches the size specified in audit_size, and adds a timestamp to the backup file name. However, if no audit events are written to the audit log file in the specified interval, CA Access Control does not back up the file after the interval elapses.

  Note: CA Access Control counts the specified interval from the time that it creates the first audit log file, and backs up the file at midnight on the appropriate day.

Example: The configuration setting has a value of weekly and CA Access Control creates the audit log file at 9:00 a.m. Friday 1 April. Many audit events occur this week and the audit log file exceeds the audit_size configuration setting on Monday 4 April. CA Access Control backs up the audit log file on 4 April and adds a timestamp to the backup file name. A week after the audit log file was first created, at midnight Friday 8 April, CA Access Control again backs up the audit log file and adds a timestamp to the backup file name.

Limits: You must specify values in all uppercase or all lowercase.

Default: yes

error_back

The name of the CA Access Control error backup file.

Default: *ACInstallDir*\log\seos.error.bak

error_group

The group that can read the error log files.

If this value is set to none, only Administrators can read the file.

Default: none

error_log

The name of the CA Access Control error log file. When this file reaches the size specified in error_size, CA Access Control closes the file, renames it with the name in error_back, and creates a new error log. Only CA Access Control can write to this file.

Default: *ACInstallDir*\log\seos.error

error_size

The maximum size, in KB, of the CA Access Control error log file.

Default: 50

irecorder_audit

Specifies whether the IR API library routes audit events of existing PMDs in addition to the local security service audit events.

all - routes audit events of Policy Models in addition to the local security service audit events.

localhost - routes audit events of the local security service only.

Default: all

SendAuditToNativeChannel

(Windows 2008 only) Specifies whether seosd sends audit events to the Windows 2008 event log channel for CA Access Control (1).

Default: 0 (no)

SendAuditToNativeLog

Specifies whether seosd sends audit events to the Windows event log (1).

Default: 0 (no)

message

CA Access Control maintains messaging settings it uses under the following key:

`HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\message`

The message registry key contains the following registry entries:

filename

The name of the file that supplies most of the messages that appear in response to CA Access Control commands.

Default: *ACInstallDir*\Data\SeOS.msg

MessagesDirectory

Specifies the location of the CA Access Control messages file.

Default: *ACInstallDir*\Data\Messages

OS_user

CA Access Control maintains enterprise user settings it uses under the following key:

`HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\OS_user`

The OS_user registry key contains the following registry entries:

create_user_in_db

Specifies whether CA Access Control creates an XUSER record for a user who is not defined to CA Access Control, when that user logs in.

Note: This setting applies only if you use enterprise users (osuser_enabled is set to 1).

Valid values are:

0 - CA Access Control does not automatically create an XUSER record.

1 - CA Access Control automatically creates an XUSER record

Default: 1

osuser_enabled

Specifies whether enterprise users and groups are enabled.

Valid values are:

0 - The use of enterprise users and groups is disabled.

1 - The use of enterprise users and groups is enabled.

Default: 1

passwd

CA Access Control maintains password settings it uses under the following key:

`HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\passwd`

The passwd registry key contains the following registry entries:

DefaultPgroup

Internal use only.

Default: other

Dictionary

Defines the full pathname of the file containing the words that *cannot* be used as passwords.

Note: To use this file, you must set the dictionary format password rule (use_dbdict) to *file* and set UseDict setting to *yes*. If the dictionary format is set to *db*, passwords that cannot be used are taken from the CA Access Control database and this setting is ignored.

Default: *ACInstallDir*\data\words

EnforceViaEtrust

Specifies whether to enforce updating or creating users' passwords through CA Access Control only.

Default: 0 (do not have to use CA Access Control)

PasswordTimeOut

Defines the maximum number of milliseconds that the CA Access Control password filter waits for authorization response.

Default: 4000

PasswordTimeOutAnswer

Specifies the answer to send back to the LSA if the authorization process does not respond in the time-out given.

If you set this to 0, the password change is refused. If you set this to 1, the password change is approved.

Default: 0

UseDict

Specifies whether to use the dictionary file (set with the Dictionary setting) when verifying a password.

Note: To use the dictionary file, you must also set the dictionary format password rule (use_dbdict) to *file*. If the dictionary format is set to *db*, passwords that cannot be used are taken from the CA Access Control database and this setting is ignored.

Default: no

Pmd

CA Access Control maintains generic Policy Model settings it uses under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Pmd
```

The Pmd registry key contains the following registry entries:

__pmd_backup_directory__

Defines the directory that CA Access Control uses to store Policy Model backups. CA Access Control stores each PMD backup in a subdirectory named *pmd_name*.

Default: *ACInstallDir*\Data\policies_backup

_Pmd_directory_

Defines the directory in which PMDB database files are located.

Default: *ACInstallDir*\Data

ClientOperationTimeout

Defines the number of seconds a Policy Model client on this computer waits for a response from the Policy Model. If the Policy Model does not respond within this time frame, the Policy Model client assumes that the Policy Model is nonresponsive.

Default: 60

MaximumPolicyModels

Defines the maximum number of policy models you can create.

Default: 16

SendAuditToNativeLog

Specifies if CA Access Control sends Policy Model audit events to the Windows event log.

Values: 0—do not send audit events to the Windows event log, 1—send audit events to the Windows event log.

Default: 0

ShutdownWaitingTimeout

Defines the number of milliseconds a Policy Model on this computer waits for its components to shut down gracefully. If Policy Model components do not shut down gracefully within this time frame, the Policy Model forces them to shut down.

Default: 60000 (1 minute)

TCPReceiveTimeout

Defines the number of seconds a Policy Model on this computer waits for a response from its subscribers. If a Policy Model subscriber does not respond within this time frame, the Policy Model closes its connection to it.

Default: 60

<PMDB_Name>

CA Access Control maintains specific Policy Model settings it uses under the following key:

`HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Pmd\`*`PMDB_Name`*

Each Pmd*PMDB_Name* registry key contains the following registry entries:

_Min_Retries

Defines the number of failed attempts the Policy Model makes to connect to a subscriber before it considers it unavailable.

Default: 4

_Retry_Timeout

Defines the time, in minutes, that the Policy Model waits before trying to resend an update to an unavailable subscriber, after the minimum number of attempts specified in _Min_Retries has been made.

Default: 30

_Shutoff_Time_

Obsolete.

Active_Policy

Defines the Policy Model name.

Always_Propagate

Specifies whether the Policy Model propagates commands when there is an error. By default, the Policy Model always sends commands for propagation. If you set this to *no* the Policy Model will not send command when there is an error.

Default: Yes

Auto_Truncate

Specifies if sepmd truncates the updates file if you execute sepmd -t without specifying either auto or the offset.

Values: Yes—sepmd automatically truncates the update file if no sepmd -t parameter is specified, No—sepmd does not truncate the update file if no sepmd -t parameter is specified

Default: Yes

Filter

Defines the full pathname of the filter file for the update file.

No default.

force_auto_truncate

Specifies whether CA Access Control truncates the update file even if there are no subscribers to the Policy Model.

You can truncate the update file manually (sepmd -t), and CA Access Control also truncates the file automatically based on a separate configuration setting (trigger_auto_truncate) that defines the event that triggers automatic truncation.

Note: If all subscribers to the Policy Model are "Out of sync", the Policy Model effectively has no subscribers.

Default: Yes

Parent_Pmd

Defines the names of parent PMDBs from which this Policy Model accepts updates.

No default.

trigger_auto_truncate

Defines the size of the Policy Model update file, in megabytes, that triggers an automatic truncating of the update file.

If you set this entry to 0, CA Access Control uses the hard-coded default value (100 MB). If you use a value that is greater than the upper limit, CA Access Control uses the upper limit value.

Type: REG_DWORD

Limits: 1 - 2000 MB

Default (DMS__ and DH__WRITER): 1024 MB

Default (all other PMDBs): 100 MB

UseEncryption

Specifies whether update information that is saved to the updates.dat file is encrypted.

Values: 0—Do not encrypt the updates.dat file, 1—encrypt the updates.dat file

Default: 0

logmgr

CA Access Control maintains specific Policy Model log settings it uses under the following key:

`HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Pmd\`*`PMDB_Name`*`\logmgr`

Each Pmd*PMDB_Name*\logmgr registry key contains the following registry entries:

audit_back

Defines the name of the Policy Model audit backup file. Only CA Access Control can write to this file.

Default: pmd_audit.bak

audit_group

Defines the group that can read the audit logs.

Default: Computer Associates

audit_log

Defines the name of the Policy Model audit log file. When this file reaches the size specified in audit_size, CA Access Control closes the file, renames it with the name set in audit_back, and creates a new audit log. Only CA Access Control can write to this file.

Default: pmd.audit

audit_size

Defines the maximum size, in KB, of the Policy Model audit log file. Do not specify a value that is less than 50 KB.

Default: 1024

error_back

Defines the name of the Policy Model error backup file.

Default: pmd_error.back

error_group

Defines the group that can read the error log files.

If this value is set to *none*, only Administrators can read the file.

Default: none

error_log

Specifies the name of the Policy Model error log file. When this file reaches the size specified in error_size, CA Access Control closes the file, renames it with the name in error_back, and creates a new error log. Only CA Access Control can write to this file.

Default: pmd.error

error_size

Defines the maximum size, in KB, of the CA Access Control error log file.

Default: 1024

<DMS_Name>

CA Access Control maintains specific DMS settings it uses under the following key:

`HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Pmd\`*`DMS_Name`*

The Pmd*DMS_Name* registry key contains the following registry entries:

_Min_Retries

Defines the number of failed attempts the Policy Model makes to connect to a subscriber before it considers it unavailable.

Default: 4

_Retry_Timeout

Defines the time, in minutes, that the Policy Model waits before trying to resend an update to an unavailable subscriber, after the minimum number of attempts specified in _Min_Retries has been made.

Default: 30

_Shutoff_Time_

Obsolete.

Active_Policy

Defines the Policy Model name.

Always_Propagate

Specifies whether the Policy Model propagates commands when there is an error. By default, the Policy Model always sends commands for propagation. If you set this to *no* the Policy Model will not send command when there is an error.

Default: Yes

Auto_Truncate

Specifies if sepmd truncates the updates file if you execute sepmd -t without specifying either auto or the offset.

Values: Yes—sepmd automatically truncates the update file if no sepmd -t parameter is specified, No—sepmd does not truncate the update file if no sepmd -t parameter is specified

Default: Yes

Filter

Defines the full pathname of the filter file for the update file.

No default.

force_auto_truncate

Specifies whether CA Access Control truncates the update file even if there are no subscribers to the Policy Model.

You can truncate the update file manually (sepmd -t), and CA Access Control also truncates the file automatically based on a separate configuration setting (trigger_auto_truncate) that defines the event that triggers automatic truncation.

Note: If all subscribers to the Policy Model are "Out of sync", the Policy Model effectively has no subscribers.

Default: Yes

Parent_Pmd

Defines the names of parent PMDBs from which this Policy Model accepts updates.

No default.

trigger_auto_truncate

Defines the size of the Policy Model update file, in megabytes, that triggers an automatic truncating of the update file.

If you set this entry to 0, CA Access Control uses the hard-coded default value (100 MB). If you use a value that is greater than the upper limit, CA Access Control uses the upper limit value.

Type: REG_DWORD

Limits: 1 - 2000 MB

Default (DMS__ and DH__WRITER): 1024 MB

Default (all other PMDBs): 100 MB

UseEncryption

Specifies whether update information that is saved to the updates.dat file is encrypted.

Values: 0—Do not encrypt the updates.dat file, 1—encrypt the updates.dat file

Default: 0

endpoint_management

CA Access Control maintains specific DMS Endpoint Management settings it uses under the following key:

`HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Pmd\`*`DMS_NAME`*`\endpoint_management`

dmsmgr defines the registry values in this key when it creates a DMS. This key is not defined if a DMS does not exist on the host.

The Pmd*DMS_Name*\endpoint_management registry key contains the following registry entries:

AutoSync

Specifies to automatically synchronize the Distribution Host with the Message Queue server.

Limits: 0,1

Default: 0 (disabled)

commands_to_exec_before_sleep

Specifies the number of endpoint commands that the DMS executes in a loop before sleeping.

Default: 10

debug_mode

Specifies if CA Access Control writes debug messages to the endpoint_management.log file in the DMS directory (1).

Limits*:* 0,1

Default*:* 0 (debugging is disabled)

Note: The log file is located at *DMSInstallDirectory*\endpoint_management.log

operation_mode

Specifies whether central (DMS) endpoint management through the CA Access Control Message Queue is enabled.

Limits: 0,1

Default: 1 (enabled)

sleep_between_exec_commands

Specifies the length of time, in milliseconds, that the DMS sleeps. When the DMS wakes it performs the number of endpoint commands specified in the commands_to_exec_before_sleep registry value.

Default: 100

policyfetcher

CA Access Control maintains policyfetcher service settings it uses under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\policyfetcher
```

The policyfetcher registry key contains the following registry entries:

check_deployment_tasks

Defines how often, in seconds, policyfetcher checks for new deployment tasks (DEPLOYMENT resources) on the Distribution Host.

Default: 3600 (every 10 minutes)

deploy_timeout

Defines the number of seconds policyfetcher waits for a deployment or undeployment task to complete on the endpoint.

Default: 900

devcalc_command

Defines the selang command that policyfetcher uses to run the deviation calculation.

Default: start DEVCALC params(-nonotify)

Example: start DEVCALC params(-nonotify -precise)

dh_command_retry_interval

Defines the number of seconds between each DH notification command retry.

Default: 60

endpoint_heartbeat

Defines the frequency at which policyfetcher sends a heartbeat to the Distribution Host (DH). The frequency is a factor of the check_deployment_task setting, and determines how many times policyfetcher checks deployment tasks before it sends a heartbeat. For example, if check_deployment_task is set to the default 600 seconds (10 minutes) and you set this to 6, policyfetcher sends a heartbeat every 3600 seconds (1 hour).

After sending the heartbeat, the policyfetcher also runs the deviation calculator (start devcalc command) and then waits 60 seconds for the deviation calculation to complete. After 60 seconds, policyfetcher continues to check that local endpoint information is identical to DH information.

Default: 6

max_deployment_errors

Defines the maximum number of deployment errors that the endpoint sends to the DMS.

Default: 10

max_dh_command_retry

Defines the maximum number of times policyfetcher retries to get update notifications from DH before giving up.

Default: 10

max_dh_retry_cycles

Defines the maximum number of cycles policyfetcher retries to get update notifications from production DHs before moving to disaster recovery DHs.

Default: 5

policy_verification

Specifies whether policyfetcher verifies new deployment tasks on a backup CA Access Control database before executing the tasks.

Valid values:

1 - Run policy verification

0 - Disable policy verification

Default: 0

policyfetcher_enabled

Specifies whether to run the policyfetcher service.

Valid values:

1 - Run policyfetcher

0 - Disable policyfetcher

Default: 0

PUPMAgent

CA Access Control maintains the Privileged User Password Management Agent settings it uses under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\PUPMAgent
```

The Privileged User Password ManagementAgent registry key contains the following registry entries:

EnableLogonIntegration

Specifies that terminal integration is enabled.

Limits: 0, terminal integration is disabled; 1, terminal integration is enabled.

Default: 1

EnableRunAsInterface

Specifies whether the Privileged User Password Management Agent is prompted for the target user password.

Limits: 0, the Privileged User Password Management Agent is not installed, 1 the Privileged User Password Management Agent is installed.

Default: 1

InterfaceName

Defines the interface name that the Privileged User Password Management Agent uses to handle requests.

Default: PUPMAgentInterface

OperationMode

Specifies the Privileged User Password Management Agent work mode.

Limits: 0, the Privileged User Password Management Agent is disabled and not running; 1, the Privileged User Password Management Agent is enabled, running but not logging data to trace files; 2, the Privileged User Password Management Agent is enabled, running, and logging data to trace files.

Default: 0

ProcessArgumentsReplacement

Specifies whether the Privileged User Password Management Agent support Process Arguments Replacement.

Limits: 0,1

Default: 0

Note: If choose to support Process Arguments Replacement, that is, you set the value of this registry entry to 1, you must also enable the CMD Plugin. To enable the CMD Plugin, set the following registry entry to 1:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Instrumentation\plugins\CMDPlg\OperationMode

Report

CA Access Control maintains sereport settings it uses under the following key:

`HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Reports`

The Reports registry key does not contain any registry entries. It contains registry subkeys for every report sereport produces.

Note: For information about registry entries for each of the reports sereport produces, see the sereport utility (see page 201).

colors

CA Access Control maintains sereport style settings it uses under the following key:

`HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Reports\colors`

The Reports\colors registry key contains the following registry entries:

background

Internal use only.

This key should remain unchanged.

class_title

Defines the color of the report's class_title.

Default: green

logo

Defines the full pathname to the logo file.

Default: *ACInstallDir*\data\logo.jpg

title

Defines the color of the report's title.

Default: midnightblue

ReportAgent Key—Registry Settings

CA Access Control maintains Report Agent settings it uses under the following key:

`HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\ReportAgent`

The ReportAgent registry key contains the following registry entries:

audit_enabled

Specifies whether you want to send endpoint audit data to the Distribution Server.

Values: 0—no; **1**—yes

Default: 0

audit_filter

Defines the full pathname to the file that contains filtering rules for audit records that the Report Agent routes to an external source (such as CA Enterprise Log Manager). This file determines which records the Report Agent routes.

Default: *ACInstallDir*\Data\AuditRouteFlt.cfg

audit_queue

Defines the name of the queue to which the Report Agent sends endpoint audit data.

Default: queue/audit

audit_read_chunk

Defines the maximal audit records the Report Agent tries to collect in a single read of the audit files.

Limits: A positive integer.

Default: 300

audit_send_chunk

Defines the maximal audit records that the Report Agent sends to the Distribution Server in each connection. When the number of audit records the Report Agent collects reaches this number it sends these records to the Distribution Server.

Limits: A positive integer

Default: 1800

audit_sleep

Define the length of time the Report Agent sleeps between generating audit reports.

Limits: A positive integer representing a number of seconds.

Default: 10

audit_timeout

Defines the cycle at which the Report Agent must send endpoint audit data to the Distribution Server. If this amount of time passes from the last send, the Report Agent sends audit data to the Distribution Server even if the number of records it collected is less than the audit_send_chunk value.

Limits: A positive integer representing a number of seconds.

Default: 300

interval

Defines the interval, in minutes, at which CA Access Control generates and sends reports to the Distribution Server.

The *schedule* setting defines the interval start time and the days it operates on. If the Report Agent starts later than a scheduled occurrence, it sends a report at the next calculated interval (from the schedule) and then at the defined intervals after that on scheduled days.

Example: If you have schedule=8:30@Mon,Tue,Wed and interval=5 and the Report Agent loads on Tuesday at 8:47 am, the Report Agent generates and sends a report at 8:50 am. This is the earliest cycle calculated from the scheduled start using the 5 minute interval.

Values: 0—No interval (use scheduled occurrences only); *positive integer*—number of minutes to use as interval

Default: 0

reportagent_enabled

Specifies whether reporting is enabled (1) on the local computer.

Default: 0

schedule

Defines when reports are generated and sent to the Distribution Server.

You specify this setting in the following format: time@day[,day2][...]

For example, "19:22@Sun,Mon" generates reports every Sunday and Monday at 7:22 pm.

Default: 00:00@Sun,Mon,Tue,Wed,Thu,Fri,Sat

send_queue

Defines the name of the reporting queue on the Distribution Server to which the Report Agent sends snapshots of the local database and any PMDBs.

Default: queue/snapshots

restart_enabled

Specifies restart of the ReportAgent daemon. Specify 1 to enable the restart.

Default: 0

More information:

auditrouteflt.cfg File—Filter Audit Records Routing (see page 396)

SeOSD Key—Registry Settings

CA Access Control maintains generic settings it uses under the following key:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\SeOSD

The SeOSD registry key contains the following registry entries:

AuditCollectorInterfaceName

Defines the pipe name which functions as an audit interface between the audit collector component (within seosd) and the different clients of the audit collector (kernel).

Default: AuditCollector

AuditServerCacheSize

Defines the size of the audit cache, in number of entries.

Default: 1024

CreateNewClasses

Specifies whether you can add new classes, created with the seclassadm utility, to a CA Access Control database.

Default: yes

CreateNewProps

Determines whether you can add new properties, created with the sepropadm utility, to a CA Access Control database.

Default: yes

dbdir

The directory in which the CA Access Control database is located.

Default: *ACInstallDir*\data\seosdb

DefLookupThreads

Defines the number of threads that CA Access Control can use to resolve SIDs into account names.

Default: 5

DefLookupTimeout

Defines the timeout, in milliseconds, before CA Access Control stops trying to resolve an SID into an account name.

Default: 2000

domain_names

The list of name suffixes used for matching purposes.

CA Access Control appends these suffixes to short host names to create long, fully qualified host names. These names can be authorized in the relevant HOST, CONNECT, or TERMINAL classes. To identify a full name, CA Access Control tries to append domain names in the domain_names list to the short name for authorization purposes. For class HOSTNP, CA Access Control matches all domain names (listed in this registry) with pattern to resolve into real IP addresses.

No default.

EnablePolicyCache

This value controls whether the authorization engine uses cached records or records directly from the database.

Valid values:

no - Authorization engine uses database records.

yes - Authorization engine uses cache records.

Default: no

EnvVarResolvingMode

The method of resolving embedded environment variables (for objects in the FILE, SECFILE, PROGRAM, PROCESS, SPECIALPGM, TERMINAL, or USER classes). For example:

newfile %SystemRoot%\temp.txt.

If you select 0, CA Access Control tries to resolve all environment variables, an error message is issued to the user, and the object is not created.

If you select 1, CA Access Control tries to resolve all environment variables, a warning message is issued to the user, and the object is created.

If you select 2, CA Access Control tries to resolve all environment variables and the object is created with no messages.

If you select 3, CA Access Control does not try to resolve environment variables.

Note: The PMDB assumes that there are no environment variables, so resolving is never tried.

Default: 2

GeneralInterceptionMode

Specifies whether to use Full Enforcement mode (0) or Audit Only mode (1).

Default: 0

GraceCountForMessage

Defines the number of remaining grace logins at which the Change Password dialog appears.

Default: 0

HostResolutionMode

Specifies the method CA Access Control uses to resolve host names.

Values:

0—HOST resolution is synchronous (current behavior).

1—HOST resolution is asynchronous (with 'Event Log' reporting)

The effects of this setting are:

- Control is returned to selang immediately.
- If a HOST record cannot be resolved, a selang message is not displayed (same as 0).
- A notification message is written into the 'Event Log'.

2—HOST resolution is asynchronous (without 'Event Log' reporting).

Same as '1' with the exception that notification messages are *not* written anywhere.

Default: 0

HostResolutionRenewal

The time for internal cache refresh. The network interception authorization events use the registry value.

Default: 30000

HostResolutionTimeout

The time the authorization engine waits for reverse IP lookup requests, upon network interception event.

Default: 2000

LogonTimeOut

Defines the time in milliseconds CA Access Control waits for transactions with the sub authentication DLL (eACSubAuth.dll) before giving up. When this time passes, CA Access Control replies with the value set in LogonTimeOutAnswer.

Default: 4000

LogonTimeOutAnswer

Defines the logon answer to the operating system when the LogonTimeOut setting elapses without an answer from CA Access Control.

Default: 1 (true)

MaximumDiscreteFILELimit

The number of discrete FILE records you can create in the CA Access Control database.

The minimum value is default; if a user sets this value to be less than the default, CA Access Control acts as if a minimum were set.

Default: 4096

MaximumGenericFILELimit

The number of generic FILE records (name pattern-based records) you can create in the CA Access Control database.

The minimum value is default; if a user sets this value to be less than the default, CA Access Control acts as if a minimum were set.

Default: 512

ProcessCreationNotificationMode

Specifies whether to intercept process creation and notify seosd either using kernel or instrumentation mode.

Type: REG_DWORD

Values:

0—Process creation is performed using kernel module

1—Process creation is performed using instrumentation module

Default: 0

Note: If you set the key to 1, CA Access Control intercepts process creation through the Windows API only.

RebuildSuspiciousDatabase

This value is addressed only if database was not properly closed on previous session.

If the value is set to 0, the database is verified in a heuristic procedure for correctness (during startup). If the check finds a problem in the database, the database is rebuilt.

If the value is set to 1, the heuristic procedure check function is skipped. The database is rebuilt according to the database integrity check.

Default: 1

RefreshIPInterval

The time (in minutes) between consecutive automatic IP refresh requests.

If the value is set to 0, IP refreshes are not automatically performed. If you use a value from 1 through 30, CA Access Control uses 30 minutes, which is the minimum amount of time you can set, as the value.

Note: Refresh requests can be time consuming. For more information, see the secons utility -refIP option.

Default: 0

ResponseFile

The location where the response.ini, used by eACOexist.exe utility, resides.

Default: *ACInstallDir*\data\response.ini

sim_login_timeout

Defines the timeout (in minutes) before CA Access Control removes unused simulated login user entries from the Accessor Element Entry table (ACEE).

CA Access Control performs a simulated login to create ACEE entries when it needs access to information that can be found in the ACEE.

Default: 60

SurrogateInterceptionMode

Specifies the SURROGATE class interception mode.

Type: REG_DWORD

Limits: 0 - user mode interception, CA Access Control intercepts only the impersonation requests that originate from the RunAs utility; 1 - kernel mode interception, CA Access Control intercepts all impersonation requests.

Default: 0

SusrauthReadParamsSec

Defines how often trace parameters are updated.

Default: 30

SusrauthTraceDbgEnable

Specifies whether tracing into DbgView or kernel debugger is enabled (1).

Default: 0

SusrauthTraceFileEnable

Specifies whether tracing into a trace file (SusrauthTraceFileName) is enabled (1).

Default: 0

SusrauthTraceFileName

Defines the full pathname to the trace file.

No default

TerminalSearchOrder

Specifies how the authorization engine determines which TERMINAL record it verifies during the authorization process.

Values:

name—Authorization engine first looks for a TERMINAL record by name and if one is not found, it looks for an IP address match.

nameonly—Authorization engine looks for a TERMINAL record by name and if one is not found, ceases searching. It ignores TERMINAL records with an IP address format.

IP—Authorization engine first looks for a TERMINAL record by IP address and if one is not found, it looks for a name match.

Note: TERMINAL class supports generic rules defined by wildcards (IP address or host name pattern match). Generic rules are always verified after specific (full-name) rules. For example, if you set this to IP, seosd looks for a TERMINAL resource in the following order: complete IP address match, complete host name match, IP address pattern match, host name pattern match.

Default: nameonly

TermSrvTimeout

Specifies the timeout (in milliseconds) that the authorization engine waits for the second consecutive login, upon a Terminal Services connection.

Default: 2000

Note: When a user logs in using a local account, CA Access Control receives two login attempt notifications: the first from the local terminal and the second from the terminal server. If the user is assigned grace login count, two login attempt are logged and reduces from the grace count. Therefore, CA Access Control does not update the grace count with the second login if the login attempt occurred within the specified timeout period.

trace_file

The name of the file to which the trace messages are sent, if trace messages are requested.

Default: *ACInstallDir*\log\seosd.trace

trace_file_type

Type of trace file.

If you do change the value of the value and a trace file exists, the existing trace file is saved with the file name extension .backup and then a new trace file is started in the format you specified.

Default: text

trace_filter

The name of the file that contains the filter data that is used to filter the trace messages. Specify the full path of the file.

Default: *ACInstallDir*\log\trcfilter.ini

trace_space_saver

The amount of free space, in KB, to be left in the file system. When the amount of free space is less than this number, CA Access Control disables the trace.

Note: Trace is never automatically enabled, even if more space becomes available at a later time.

Default: 5120

trace_to

The destination of trace messages. Set to none, file, or file,stop.

If you select none, CA Access Control does not generate trace messages.

If you select file, CA Access Control generates trace messages and sends them to the file listed in the registry trace_file as soon as CA Access Control becomes active.

If you select file,stop, CA Access Control generates trace messages during the period of service initialization. Once the service is initialized, no more trace messages are generated.

Default: file,stop

SeOSWD

CA Access Control maintains watchdog settings it uses under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\SeOSWD
```

The SeOSWD registry key contains the following registry entries:

PgmRest

Specifies the period, in seconds, after the last event and before checking programs again. The program rests to prevent system overload.

Default: 10

PgmTestInterval

The period, in seconds, between rescanning of programs.

Default: 18000

SecFileRest

Specifies the period, in seconds, after the last event and before checking secured files again. The program rests to prevent system overload.

Default: 10

SecFileTestInterval

The period, in seconds, between rescanning of secured files.

Default: 36000

STOP

CA Access Control maintains Stack Overflow Protection (STOP) settings it uses under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\STOP
```

The STOP registry key contains the following registry entries:

STOPIniFileName

Defines the full path and name of the STOP initialization file. This file contains the list of functions for which STOP is enabled.

Default: *ACInstallDir*\Data\stop.ini

STOPLearningModeEnabled

Specifies whether STOP runs in a special learning mode. In this mode, incidents are logged but always permitted. That is, a denial incident is logged appropriately, but is permitted to continue.

Default: 0 (disabled)

STOPLogFileName

Defines the full path and name of the dynamic incident database for stack overflow protection (STOP).

Default: *ACInstallDir*\Log\STOPRTEvents.dat

STOPServerTraceEnabled

Specifies whether the STOP server module has trace logging enabled.

Default: 0 (disabled)

STOPSignatureBrokerName

Defines the host name of the computer that (if defined) is used to retrieve STOP signatures database from.

No default.

STOPSignatureFileName

Defines the full path and name of the STOP signature file (a trusted incident database).

Default: *ACInstallDir*\Data\stopsignature.dat

STOPUpdateInterval

Defines the period of time, in minutes, between two consecutive attempts to update the STOP signatures database.

Default: 60

STOPZeroSnapshotBypassEnabled

Specifies whether STOP should permit incidents with a zero-size code snapshot.

Default: 0 (not permitted)

Tracer

CA Access Control maintains tracing module settings it uses under the following key:

`HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Tracer`

The Tracer registry key contains the following registry entries:

TraceCfgFile

Defines the full path of the file containing the initialized configuration settings for tracing CA Access Control modules.

Default: *ACInstallDir*\Data\tracer.ini

TraceEnabled

Specifies whether to enable the Trace mechanism.

Default: 0 (disabled)

UCTNG

CA Access Control maintains Unicenter integration settings it uses under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\UCTNG
```

The UCTNG registry key contains the following registry entries:

EvtManagerServer

Defines the name of the Unicenter TNG host.

Integration

Specifies whether to enable integration with Unicenter TNG and send audit data.

Default: 0 (do not enable integration)

uxauth Key—Registry Settings

UNIX Authentication Broker maintains Active Directory schema settings that it uses under the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\uxauth
```

UNIX Authentication Broker installs this registry key when you install the CA Access Control UNIX Attributes plug-in on an Active Directory server. This registry key is not installed as part of CA Access Control.

Note: The default attributes are for the Active Directory 2003 R2 schema.

The uxauth registry key contains the following registry entries:

group_gid_attr_name

Specifies the Active Directory attribute to which UNIX Authentication Broker maps the GID for a migrated UNIX group.

Default: gidNumber

Trace_Enabled

Specifies if tracing is enabled for the CA Access Control UNIX Attributes plug-in.

Values: 0—tracing is disabled, 1—tracing is enabled

Default: 0

user_gecos_attr_name

Specifies the Active Directory attribute to which UNIX Authentication Broker maps the gecos property for a migrated UNIX user.

Default: gecos

user_gid_attr_name

Specifies the Active Directory attribute to which UNIX Authentication Broker maps the GID for a migrated UNIX user.

Default: gidNumber

user_homedir_attr_name

Specifies the Active Directory attribute to which UNIX Authentication Broker maps the home directory property for a migrated UNIX user.

Default: unixHomeDirectory

user_loginshell_attr_name

Specifies the Active Directory attribute to which UNIX Authentication Broker maps the login shell property for a migrated UNIX user.

Default: loginShell

user_uid_attr_name

Specifies the Active Directory attribute to which UNIX Authentication Broker maps the UID of a migrated UNIX user.

Default: uidNumber

WebService

CA Access Control maintains Web Service settings it uses under the following key:

`HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\WebService`

Note: The WebService registry key and related entries are added as part of the CA Access Control Endpoint Management installation.

The WebService registry key contains the following registry entries:

auditFileCheckInterval

Defines how often, in seconds, the CA Access Control Web Service checks if the audit file size has reached the defined limit.

Default: 60

auditFileMaxSize

Defines the maximum size in KB of the CA Access Control Web Service audit log file.

When the file reaches this size, the Web Service renames the file to *"Backup_of_logFileName*" and creates a new audit log file.

Default: 20000

backLog

Defines the maximum size of the request queue the CA Access Control Web Service maintains.

Default: 101

logFileName

Defines the name of the CA Access Control Web Service audit log file name.

If you leave this value empty string (""), the Web Service sends log messages to the terminal when you run the Web Service with the -debug option.

Default: *ACServerInstallDir*\WebService\log\WebService.log

machineName

Defines the name of the computer the CA Access Control Web Service is installed on.

Default: 127.0.0.1

maxRequestsQueue

Defines the size of the global request queue of sockets.

Default: 1001

maxThreads

Defines the number of threads CA Access Control Web Service uses.

Default: 7

portNumber

Defines the port CA Access Control Web Service uses to communicate.

Default: 5248

sessionTimeOut

Defines the number of seconds before CA Access Control Web Service terminates a session when there is no operation.

Default: 601

StandAloneService

Specifies whether the CA Access Control Web Service operates as a standalone service.

If the CA Access Control Web Service operates as a standalone service, the service is not stopped or started when you use secons to stop or seosd to start CA Access Control services. Instead, you use Windows native tools to start and stop the CA Access Control Web Service.

If the CA Access Control Web Service does not operate as a standalone service, the service is stopped and started when you use secons to stop or seosd to start CA Access Control services. You cannot use Windows native tools to start and stop the CA Access Control Web Service. However, to use seosd -start to start the CA Access Control Web Service, you must define the CA Access Control Web Service in the AccessControl\AccessControlServices registry entry.

Values: 1—Operates as a standalone service; 0—Does not operate as a standalone service

Default: 1

TraceEnabled

Specifies if tracing is enabled for the CA Access Control Web Service components.

Values: 0—tracing is disabled, 1—tracing is enabled

Default: 0

Additional Registry Keys

You can also add or modify the following keys and values to change the way CA Access Control performs:

Registry Entry	Type	Description
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\drveng\Parameters\DisableFileInterception	REG_DWORD	Specifies whether the file interception hooking is disabled (relevant functions are not initialized at boot time). **Value:** 1 (disabled) **Note:** If this registry entry does not exist (the default), or is set to any value other than 1, file interception is initialized at boot time.

Registry Entry	Type	Description
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\drveng\Parameters\DisableNetworkInterception	REG_DWORD	Specifies whether network interception hooking is disabled (relevant functions are not initialized at boot time). **Value:** 1 (disabled) **Note:** If this registry entry does not exist (the default), or is set to any value other than 1, network interception is initialized at boot time.
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\drveng\Parameters\DisableProcessInterception	REG_DWORD	Specifies whether process interception hooking is disabled (relevant functions are not initialized at boot time). **Value:** 1 (disabled) **Note:** If this registry entry does not exist (the default), or is set to any value other than 1, process interception is initialized at boot time.
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\drveng\Parameters\DisableRegistryInterception	REG_DWORD	Specifies whether the registry interception hooking is disabled (relevant functions are not initialized at boot time). **Value:** 1 (disabled) **Note:** If this registry entry does not exist (the default), or is set to any value other than 1, registry interception is initialized at boot time.
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SeosDrv\Parameters\KernelBuffersSize	REG_DWORD	When the CA Access Control kernel driver (seosdrv.sys) starts, it allocates, by default, memory for its internal use, according to the following formula: number_of_buffers = amount_of_RAM For example, 256 buffers are allocated for 256 MB of RAM. Each buffer is 4096 bytes long. If you want to control the number of buffers that seos.drv allocates, create this registry key and set the value to the number of buffers to allocate. **Note:** 32 is the minimum number of buffers.
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\SeosDrv\EventMessageFile	REG_EXPAND_SZ	Defines the pathname to the seosdrv.sys driver. **Default:** %SystemRoot%\System32\drivers\seosdrv.sys
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\SeosDrv\TypesSupported	REG_DWORD	A standard Windows entry that defines the bitmask of supported event types. **Default:** 7

Registry Entry	Type	Description
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\cainstrm\parameters\DllScanList	REG_SZ	Defines a list of comma-separated DLLs (by name) that trigger injection by cainstrm.sys **Default:** No default
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\cainstrm\parameters\DllScanListRefreshPeriod	REG_DWORD	Defines the interval, in seconds, for scanning the cainstrm registry entry. **Default:** 600
HKEY_LOCAL_MACHINE\System\CCS\Services\Cainstrm\parameters\ExcludeProcess	REG_MULTI_SZ	Specifies processes by name to be excluded from native instrumentation by the driver. **Default:** none
HKEY_LOCAL_MACHINE\SYSTEM\CCS\Services\Cainstrm\Parameters	REG_DWORD	Specifies the CA Access Control low-level instrumentation policy towards .Net assemblies. **Default:** 1 (1 implies that the instrumentation of .Net assemblies is enabled).

Appendix A: Audit Log Records

This section contains the following topics:

Audit Records

Each record in the audit log contains data that is arranged in columns. Two columns (date and time stamps) are common to all types of records. The remaining columns and the data they contain depend on the type of event that triggered the creation of the audit record.

Note: The order, number, and content of columns that you see for an audit log record depend on the method you choose to view the audit log. Some fields do not display in CA Access Control Endpoint Management, seaudit output, or the detailed seaudit output. Also, if you use the seaudit utility, the options you specify may also determine the number, order, and content of the columns.

How To Identify the Event Type of an Audit Record

To understand the content of an audit record, you must first identify the event type of the audit record. This is because the data the record contains depends on the type of event that triggered the creation of the audit record.

Note: The order, number, and content of columns that you see for an audit log record depend on the method you choose to view the audit log. Some fields do not display in CA Access Control Endpoint Management, seaudit output, or the detailed seaudit output. Also, if you use the seaudit utility, the options you specify may also determine the number, order, and content of the columns.

To identify the event type of an audit record:

- If you are viewing audit records in CA Access Control Endpoint Management, the event type the audit record belongs to displays in the first column of the Audit Records Result pane.

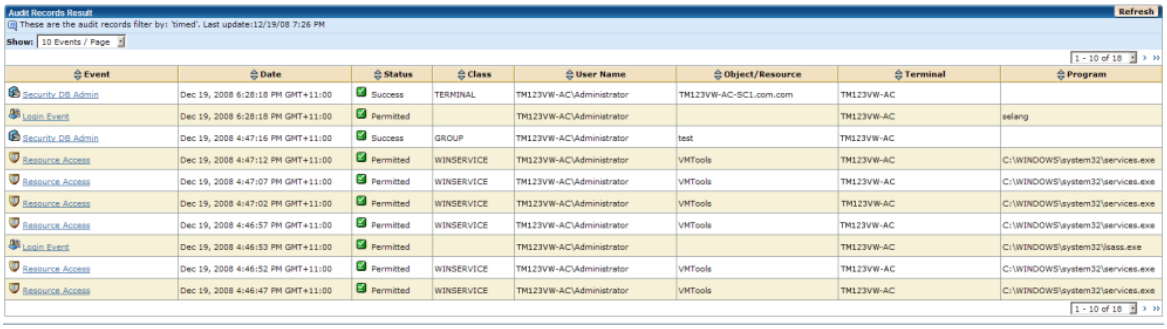
  To display more information about the audit record, click the link audit event type in the first column.

- If you are viewing audit records in seaudit output, you need to display the detailed output (-detail option) to see the event type.

Once you identify the event type, you can go on to interpret the rest of the message detail.

Example: Audit Records in CA Access Control Endpoint Management

The following image shows you how CA Access Control Endpoint Management presents audit events:



Example: Audit Records in Default seaudit Output

The following snippet of a seaudit output shows you how the seaudit utility presents audit events by default:

```
19 Dec 2008 16:46:47 P WINSERVICE  TM123VM-AC\Administrator Read    1059  2 VMTools
C:\WINDOWS\system32\services.exe TM123VM-AC
19 Dec 2008 16:46:52 P WINSERVICE  TM123VM-AC\Administrator Read    1059  2 VMTools
C:\WINDOWS\system32\services.exe TM123VM-AC
19 Dec 2008 16:46:53 P LOGIN       TM123VM-AC\Administrator   55  2 TM123VM-AC
C:\WINDOWS\system32\lsass.exe
19 Dec 2008 16:46:57 P WINSERVICE  TM123VM-AC\Administrator Read    1059  2 VMTools
C:\WINDOWS\system32\services.exe TM123VM-AC
19 Dec 2008 16:47:02 P WINSERVICE  TM123VM-AC\Administrator Read    1059  2 VMTools
C:\WINDOWS\system32\services.exe TM123VM-AC
19 Dec 2008 16:47:07 P WINSERVICE  TM123VM-AC\Administrator Read    1059  2 VMTools
C:\WINDOWS\system32\services.exe TM123VM-AC
19 Dec 2008 16:47:12 P WINSERVICE  TM123VM-AC\Administrator Read    1059  2 VMTools
C:\WINDOWS\system32\services.exe TM123VM-AC
```

```
19 Dec 2008 16:47:16 S UPDATE      GROUP     TM123VM-AC\Administrator  336  0 test
TM123VM-AC egtest audit-
19 Dec 2008 18:28:18 P LOGIN        TM123VM-AC\Administrator   55 10 TM123VM-AC
selang
19 Dec 2008 18:28:18 S UPDATE       TERMINAL   TM123VM-AC\Administrator  305  0
TM123VM-AC-SC1.ca.com TM123VM-AC er terminal TM123VM-AC-SC1.ca.com
```

The detailed seaudit output for the first message above is as follows:

```
19 Dec 2008 16:46:47 P WINSERVICE   TM123VM-AC\Administrator Read    1059  2 VMTools
C:\WINDOWS\system32\services.exe TW852VM-AC
Event type: Resource access
Status: Permitted
Class: WINSERVICE
Resource: VMTools
Access: Read
User name: TM123VM-AC\Administrator
User Logon Session ID: 00000000:05647d29
Terminal: TM123VM-AC
Program: C:\WINDOWS\system32\services.exe
Date: 19 Dec 2008
Time: 16:46
Details: Default record universal access check
Audit flags: AC database user
```

Audit Event Types

The information CA Access Control stores in the audit log is determined by the type of event it audits.

CA Access Control logs audit records for the following event types:

Login Event

Login events describe an attempt to log in to CA Access Control or a CA Access Control protected host.

Audit records in this event have the following format:

Date Time Status Event UserName SessionID Details Reason Terminal Program AuditFlags

Date

Identifies the date the event occurred.

Format: DD MMM YYYY

Note: CA Access Control Endpoint Management formats the date display according to your computer's settings.

Time

Identifies the time the event occurred.

Format: HH:MM:SS

Note: CA Access Control Endpoint Management formats the time display according to your computer's settings.

Status

Indicates the return code for the event.

Values: Can be one of:

- D (Denied)—Denied the event because of insufficient authorization.
- P (Permitted)—Permitted the event.
- W (Warning)—Permitted the event because Warning mode is set although the access request violates an access rule.

Event Type

Identifies the type of event this record belongs to.

Note: CA Access Control Endpoint Management refers to this field simply as *Event*.

User Name

Identifies the name of the accessor that performed the action that triggered this event.

User Logon Session ID

Identifies the accessor's session ID.

Note: By default this field does not appear in a non-detailed seaudit output. To display this field in a non-detailed seaudit output, specify the -sessionid option in the seaudit command.

Details

Indicates at which stage CA Access Control decided what action to take for this event.

Note: The audit record you see in a non-detailed seaudit output displays a number in this field. This number is known as the authorization stage code. In a detailed output or in CA Access Control Endpoint Management, the audit record displays the message associated with the authorization stage code. For a complete list of stage codes, run seaudit -t.

Reason

Indicates the reason that CA Access Control wrote an audit record.

Note: This field does not display in a detailed seaudit output or in CA Access Control Endpoint Management. The audit record you see in a non-detailed seaudit output displays a number in this field. This number is known as the reason code. For a complete list of reason codes, run seaudit -t.

Terminal

Identifies the name of the terminal that the accessor used to connect to the host.

Program

Identifies the name of the program that triggered the event. That is, the program that the accessor used to try to log in. For CA Access Control administration login, this is the CA Access Control module that logged in (selang, Web Service, and so on).

Audit Flags

Indicates whether the accessor is internal (CA Access Control database user) or an enterprise user.

Note: If the accessor is an enterprise user, the audit record you see in a non-detailed seaudit output displays the string "(OS user)" in this field. Otherwise, this field remains empty.

Example: Login Event Message

The following audit record was taken from a detailed seaudit output.

```
28 Oct 2008 12:15:01 P LOGIN  root  49047159:0000034b  59  2  _CRONJOB_  SBIN_CRON
Event type: Login event
Status: Permitted
User name: root
Terminal: _CRONJOB_
Program: SBIN_CRON
Date: 28 Oct 2008
Time: 12:15
Details: Resource UACC check
```

```
User Logon Session ID: 49047159:0000034b
Audit flags: AC database user
```

This audit record indicates that on October 28th 2008, at 12:15:01 user root logged in to the protected host from terminal _CRONJOB_ and ran a SBIN_CRON program. CA Access Control permitted the operation because the resource's default access permissions permit this action (authorization stage code 59—Resource UACC check). CA Access Control logged this event because the accessor's audit mode specifies that this event should be logged (reason code 2—User audit mode requires logging).

More information:

Authorization Stage Codes for Log In and Log Out Events (see page 580)
Reason Codes That Specify Why a Record Was Created (see page 614)

Logout Event

Valid on UNIX

Logout events describe an attempt to log out from CA Access Control or a CA Access Control protected host.

Note: Logout events are only supported on UNIX. CA Access Control does not actually intercept logout. Instead, it assumes logout occurs when the last process for the session terminates.

Audit records in this event have the following format:

Date Time Status Event UserName SessionID Details Reason Terminal AuditFlags

Date

Identifies the date the event occurred.

Format: DD MMM YYYY

Note: CA Access Control Endpoint Management formats the date display according to your computer's settings.

Time

Identifies the time the event occurred.

Format: HH:MM:SS

Note: CA Access Control Endpoint Management formats the time display according to your computer's settings.

Status

Indicates that a user logout occurred.

Value: O (Logout)

Event Type

Identifies the type of event this record belongs to.

Note: CA Access Control Endpoint Management refers to this field simply as *Event*.

User Name

Identifies the name of the accessor that performed the action that triggered this event.

User Logon Session ID

Identifies the accessor's session ID.

Note: By default this field does not appear in a non-detailed seaudit output. To display this field in a non-detailed seaudit output, specify the -sessionid option in the seaudit command.

Details

Indicates how the logout was detected.

Details

Indicates at which stage CA Access Control decided what action to take for this event.

Note: The audit record you see in a non-detailed seaudit output displays a number in this field. This number is known as the authorization stage code. In a detailed output or in CA Access Control Endpoint Management, the audit record displays the message associated with the authorization stage code. For a complete list of stage codes, run seaudit -t.

Reason

Indicates the reason that CA Access Control wrote an audit record.

Note: This field does not display in a detailed seaudit output or in CA Access Control Endpoint Management. The audit record you see in a non-detailed seaudit output displays a number in this field. This number is known as the reason code. For a complete list of reason codes, run seaudit -t.

Terminal

Identifies the name of the terminal that the accessor used to connect to the host.

Audit Flags

Indicates whether the accessor is internal (CA Access Control database user) or an enterprise user.

Note: If the accessor is an enterprise user, the audit record you see in a non-detailed seaudit output displays the string "(OS user)" in this field. Otherwise, this field remains empty.

Example: Logout Event Message

The following audit record was taken from a detailed seaudit output.

```
29 Jan 2009 17:23:33 0 LOGOUT       root                 49  2 computer.com
Event type: Logout
Status: Logout
User name: root
Terminal: computer.com
Date: 29 Jan 2009
Time: 17:23
Details: Logout detected after last process terminated
Audit flags: AC database user
```

This audit record indicates that on January 29th 2009, CA Access Control detected that the last session process for the user root working on the remote terminal computer.com has closed, and so assumes that the user logged out of the system (authorization stage code 49—Logout detected after last process terminated).

More information:

Authorization Stage Codes for Log In and Log Out Events (see page 580)
Reason Codes That Specify Why a Record Was Created (see page 614)

Login Account Enabled Event

Valid on UNIX

Login account enabled events describe events where serevu enables a user log in.

Audit records in this event have the following format:

Date Time Status Event UserName Details Reason Terminal Program AuditFlags

Date

Identifies the date the event occurred.

Format: DD MMM YYYY

Note: CA Access Control Endpoint Management formats the date display according to your computer's settings.

Time

Identifies the time the event occurred.

Format: HH:MM:SS

Note: CA Access Control Endpoint Management formats the time display according to your computer's settings.

Status

Indicates serevu enabled user login.

Value: E (Login enabled)

Event Type

Identifies the type of event this record belongs to.

Note: CA Access Control Endpoint Management refers to this field simply as *Event*.

User Name

Identifies the name of the accessor that performed the action that triggered this event.

Details

Indicates at which stage CA Access Control decided what action to take for this event.

Note: The audit record you see in a non-detailed seaudit output displays a number in this field. This number is known as the authorization stage code. In a detailed output or in CA Access Control Endpoint Management, the audit record displays the message associated with the authorization stage code. For a complete list of stage codes, run seaudit -t.

Reason

Indicates the reason that CA Access Control wrote an audit record.

Note: This field does not display in a detailed seaudit output or in CA Access Control Endpoint Management. The audit record you see in a non-detailed seaudit output displays a number in this field. This number is known as the reason code. For a complete list of reason codes, run seaudit -t.

Terminal

Identifies the name of the terminal that the accessor used to connect to the host.

Program

Identifies the name of the program that triggered the event.

Audit Flags

Indicates whether the accessor is internal (CA Access Control database user) or an enterprise user.

Note: If the accessor is an enterprise user, the audit record you see in a non-detailed seaudit output displays the string "(OS user)" in this field. Otherwise, this field remains empty.

Example: Login Account Enabled Event Message

The following audit record was taken from a detailed seaudit output.

```
13 Jan 2009 17:05:00 E LOGINENABLE  test1          0  5 computer.com      serevu
Event type: Login account enabled
Status: Login enabled
User name: test1
Details: Stage code 0
Terminal: computer.com
Date: 13 Jan 2009
Time: 17:05
Program: serevu
Audit flags: AC database userLogin account disable –
```

This audit record indicates that on January 13th 2009, the serevu daemon enabled user test1 to log in from the terminal computer.com. CA Access Control logged this event because the serevu daemon requested the audit (reason code 5—CA Access Control serevu utility requested auditing).

More information:

Authorization Stage Codes for Log In and Log Out Events (see page 580)
Reason Codes That Specify Why a Record Was Created (see page 614)

Login Account Disabled Event

Valid on UNIX

Login account disabled events describe events where serevu disables a user log in.

Audit records in this event have the following format:

Date Time Status Event UserName Details Reason Terminal Program AuditFlags

Date

Identifies the date the event occurred.

Format: DD MMM YYYY

Note: CA Access Control Endpoint Management formats the date display according to your computer's settings.

Time

Identifies the time the event occurred.

Format: HH:MM:SS

Note: CA Access Control Endpoint Management formats the time display according to your computer's settings.

Status

Indicates serevu disabled user login.

Value: I (Login disabled)

Event Type

Identifies the type of event this record belongs to.

Note: CA Access Control Endpoint Management refers to this field simply as *Event*.

User Name

Identifies the name of the accessor that performed the action that triggered this event.

User Logon Session ID

Identifies the accessor's session ID.

Note: By default this field does not appear in a non-detailed seaudit output. To display this field in a non-detailed seaudit output, specify the -sessionid option in the seaudit command.

Details

Indicates at which stage CA Access Control decided what action to take for this event.

Note: The audit record you see in a non-detailed seaudit output displays a number in this field. This number is known as the authorization stage code. In a detailed output or in CA Access Control Endpoint Management, the audit record displays the message associated with the authorization stage code. For a complete list of stage codes, run seaudit -t.

Reason

Indicates the reason that CA Access Control wrote an audit record.

Note: This field does not display in a detailed seaudit output or in CA Access Control Endpoint Management. The audit record you see in a non-detailed seaudit output displays a number in this field. This number is known as the reason code. For a complete list of reason codes, run seaudit -t.

Terminal

Identifies the name of the terminal that the accessor used to connect to the host.

Program

Identifies the name of the program that triggered the event.

Audit Flags

Indicates whether the accessor is internal (CA Access Control database user) or an enterprise user.

Note: If the accessor is an enterprise user, the audit record you see in a non-detailed seaudit output displays the string "(OS user)" in this field. Otherwise, this field remains empty.

Example: Login Account Disabled Event Message

The following audit record was taken from a detailed seaudit output.

```
13 Jan 2009 16:53:26 I LOGINDISABLE test1                    0  5
computer.com      serevu
Event type: Login account disable
Status: Login disabled
User name: test1
Terminal: computer.com
Date: 13 Jan 2009
Time: 16:53
Program: serevu
Details: Stage code 0
User Logon Session ID: 496b629c:00000003
Audit flags: AC database user
```

This audit record indicates that on January 13th 2009, the serevu daemon prevented user test1 from logging in from the terminal computer.com. CA Access Control logged this event because the serevu daemon requested the audit (reason code 5—CA Access Control serevu utility requested auditing).

More information:

Authorization Stage Codes for Log In and Log Out Events (see page 580)
Reason Codes That Specify Why a Record Was Created (see page 614)

Password Attempt Event

Valid on UNIX

Password attempt events describe an accessor's attempt to log in with an incorrect password.

Audit records in this event have the following format:

Date Time Status Event UserName Details Reason Terminal Program AuditFlags

Date

Identifies the date the event occurred.

Format: DD MMM YYYY

Note: CA Access Control Endpoint Management formats the date display according to your computer's settings.

Time

Identifies the time the event occurred.

Format: HH:MM:SS

Note: CA Access Control Endpoint Management formats the time display according to your computer's settings.

Status

Indicates an incorrect password attempt.

Value: A (Password attempt)

Event Type

Identifies the type of event this record belongs to.

Note: CA Access Control Endpoint Management refers to this field simply as *Event*.

User Name

Identifies the name of the accessor that performed the action that triggered this event.

Details

Indicates at which stage CA Access Control decided what action to take for this event.

Note: The audit record you see in a non-detailed seaudit output displays a number in this field. This number is known as the authorization stage code. In a detailed output or in CA Access Control Endpoint Management, the audit record displays the message associated with the authorization stage code. For a complete list of stage codes, run seaudit -t.

Reason

Indicates the reason that CA Access Control wrote an audit record.

Note: This field does not display in a detailed seaudit output or in CA Access Control Endpoint Management. The audit record you see in a non-detailed seaudit output displays a number in this field. This number is known as the reason code. For a complete list of reason codes, run seaudit -t.

Terminal

Identifies the name of the terminal that the accessor used to connect to the host.

Program

Identifies the name of the program that triggered the event.

Audit Flags

Indicates whether the accessor is internal (CA Access Control database user) or an enterprise user.

Note: If the accessor is an enterprise user, the audit record you see in a non-detailed seaudit output displays the string "(OS user)" in this field. Otherwise, this field remains empty.

Example: Password Attempt Event Message

The following audit record was taken from a detailed seaudit output.

```
13 Jan 2009 16:21:12 A LOGIN        admin                 17  8
localhost.localdomain login
Event type: Password attempt
Status: Password attempt
User name: admin
Terminal: localhost.localdomain
Date: 13 Jan 2009
Time: 16:21
Program: login
```

```
Details: Attempt rejected by the native environment
Audit flags: AC database user
```

This audit record indicates that on January 13th 2009, the user admin attempted to change his account's password. The attempt was rejected by the native environment because of a login failure (authorization stage code 17—attempt rejected by the native environment). The pam _seos module logged this event (reason code 8—CA Access Control pam support UNIX failed login).

More information:

Authorization Stage Codes for Log In and Log Out Events (see page 580)
Reason Codes That Specify Why a Record Was Created (see page 614)

Resource Access Event

Resource access events describe access attempts to resources, for example, FILE, TERMINAL, PROGRAM, and more. The audit record data in this event can appear in other records, for example, a LOGIN event when an accessor attempts to access a TERMINAL resource. Although the event record in this case is of the LOGIN type, the audit record data that appears in the record is one of the Resource Access Event messages.

Audit records in this event have the following format:

Date Time Status Class UserName SessionID Access Details Reason Resource Program Terminal EffectiveUserName AuditFlags

Date

Identifies the date the event occurred.

Format: DD MMM YYYY

Note: CA Access Control Endpoint Management formats the date display according to your computer's settings.

Time

Identifies the time the event occurred.

Format: HH:MM:SS

Note: CA Access Control Endpoint Management formats the time display according to your computer's settings.

Status

Indicates the return code for the event.

Values: Can be one of:

- D (Denied)—Denied the event because of insufficient authorization.
- P (Permitted)—Permitted the event.
- W (Warning)—Permitted the event because Warning mode is set although the access request violates an access rule.
- N (Notify)—Permitted the event and notifies that an attempt to access a permitted resource occurred.
- F (Failed)—Permitted, but the Operating System command failed.

Class

Identifies the class that the resource being accessed belongs to.

User Name

Identifies the name of the accessor that performed the action that triggered this event.

User Logon Session ID

Identifies the accessor's session ID.

Note: By default this field does not appear in a non-detailed seaudit output. To display this field in a non-detailed seaudit output, specify the -sessionid option in the seaudit command.

Access

Identifies the type of attempted access that triggered this event.

Example: Read

Note: Access values depend on the class the intercepted resource belongs to. For more information on the access authority for each class, see the *selang Reference Guide*.

Details

Indicates at which stage CA Access Control decided what action to take for this event.

Note: The audit record you see in a non-detailed seaudit output displays a number in this field. This number is known as the authorization stage code. In a detailed output or in CA Access Control Endpoint Management, the audit record displays the message associated with the authorization stage code. For a complete list of stage codes, run seaudit -t.

Reason

Indicates the reason that CA Access Control wrote an audit record.

Note: This field does not display in a detailed seaudit output or in CA Access Control Endpoint Management. The audit record you see in a non-detailed seaudit output displays a number in this field. This number is known as the reason code. For a complete list of reason codes, run seaudit -t.

Resource

Identifies the name of the actual resource that is being accessed or updated.

Program

Identifies the name of the program that triggered the event. That is, the program that the accessor used to try to access the resource.

Terminal

Identifies the name of the terminal that the accessor used to connect to the host. (UNIX only.)

Effective User Name

Identifies the name of the native OS effective user that triggered this event. This is different from the user name if the user substitutes (surrogates) to a different user or runs a setuid program.

Audit Flags

Indicates whether the accessor is internal (CA Access Control database user) or an enterprise user.

Note: If the accessor is an enterprise user, the audit record you see in a non-detailed seaudit output displays the string "(OS user)" in this field. Otherwise, this field remains empty.

Example: Resource Access Event Message

The following audit record is taken from a detailed seaudit output.

```
18 Nov 2008 15:23:56 D FILE       admabc  4922ae61:00000132 Read      69  3 /tmp/one
/usr/local/bin/tcsh  localhost      admabc
Event type: Resource access
Status: Denied
Class: FILE
Resource: /tmp/one
Access: Read
User name: admabc
Terminal: localhost
Program: /usr/local/bin/tcsh
Date: 18 Nov 2008
Time: 15:23
```

```
Details: No Step that allowed access
User Logon Session ID: 4922ae61:00000132
Audit flags: AC database user
Effective user name: admabc
```

This audit record indicates that on November 18th 2008, at 15:23:56 the user admabc used UNIX tcsh shell program from the local computer to try and read the protected /tmp/one file resource. CA Access Control denied the operation because there are no rules in the database that authorize this type of access (authorization stage code 69—No step that allowed access). CA Access Control logged this event because the resource's audit mode specifies that this event should be logged (reason code 3—Resource audit mode required logging).

More information:

Authorization Stage Codes for Resource Access Events (see page 583)
Reason Codes That Specify Why a Record Was Created (see page 614)

Untrust Message Event

Untrust events describe warning messages that the CA Access Control Watchdog generates for events.

Audit records in this event have the following format:

Date Time Status Class Module Details MessageID/errno File

Date

Identifies the date the event occurred.

Format: DD MMM YYYY

Note: CA Access Control Endpoint Management formats the date display according to your computer's settings.

Time

Identifies the time the event occurred.

Format: HH:MM:SS

Note: CA Access Control Endpoint Management formats the time display according to your computer's settings.

Status

Indicates untrust occurred.

Value: U (Untrust)

Class

Identifies the CA Access Control class that the resource that triggered the watchdog message belongs to.

Values: PROGRAM or SECFILE

Module Name

Displays the name of the CA Access Control Watchdog.

Value: seoswd

Details

Indicates why the untrust event occurred.

Note: The audit record you see in a non-detailed seaudit output displays a number in this field. This number is known as the untrust reason code. In a detailed output or in CA Access Control Endpoint Management, the audit record displays the message associated with the untrust reason code. For a complete list of password quality codes, run seaudit -t.

Message ID

(UNIX only) Indicates the reason CA Access Control untrusted the PROGRAM or SECFILE.

Note: The audit record you see in a non-detailed seaudit output displays a number in this field. This number is known as the status code and does not show in a detailed output or in CA Access Control Endpoint Management. To understand what the status code means, run seaudit -Stat *untrust_code*. This field displays only if the authorization stage code is 1. In all other cases, the errno field displays instead.

errno

Indicates the return value of the errno variable (the error code for the error condition).

Values: can be one of:

0—No error. This value is returned only if the authorization stage code is 1. In this case, the errno field is not displayed and the Message ID field displays instead.

errno—A non-zero integer that is the error.

Note: To find out the meaning for the error, on UNIX, see /usr/include/errno.h or /usr/include/sys/errno.h file on the local computer. On Windows, enter the following command on the local computer: net helpmsg *errno*

File

Identifies the full pathname of the protected resource that triggered the Watchdog message.

Example: Untrust Message Event Message

The following audit record was taken from a detailed seaudit output.

```
18 Nov 2008 14:01:18 U PROGRAM      seoswd                1 11776 /tmp/testsuid
Event type: Untrust message
Class: PROGRAM
Module name: seoswd
Message ID: 11776
Date: 18 Nov 2008
Time: 14:01
File: /tmp/testsuid
Details: Stat information changed on file system
Audit flags: AC database user
```

This audit record indicates that on November 15th 2008 the Watchdog marked the program /tmp/testsuid as untrusted (U). The program was untrusted because the file status information was modified (untrust reason code 1—File information changed on file system).

Example: Use seaudit -Stat to See Why a Program Was Untrusted (UNIX)

The following seaudit -Stat output shows you how you can get more detailed information about the Watchdog message ID that an audit record mentions.

```
# seaudit -Stat 11776
CA Access Control seaudit v12.01.00.45 - Audit log lister
Copyright (c) 2008 CA. All rights reserved.

The MODE of the file was changed
The INODE of the file was changed
The SIZE of the file was changed
The MTIME of the file was changed
```

Running the seaduit -Stat command with the message ID, displays a list of changes to the file. In this example, the MODE, INODE, SIZE, and MTIME of the file changed. As a result CA Access Control marked this file as an untrusted file.

More information:

Authorization Stage Codes for Untrust Message Events (see page 593)
Reason Codes That Specify Why a Record Was Created (see page 614)

Inbound Network Connection Event

Inbound network connection events indicate incoming traffic to the protected host. Inbound network events are audited in two forms (according to the class activation in the local database). Both audit event types contain identical information but in different view. For example, one audit event contains HOST as the class name while the other event displays TCP as the class name.

Audit records in this event have the following format:

Date Time Status Event Service Details Reason Host Program

Date

Identifies the date the event occurred.

Format: DD MMM YYYY

Note: CA Access Control Endpoint Management formats the date display according to your computer's settings.

Time

Identifies the time the event occurred.

Format: HH:MM:SS

Note: CA Access Control Endpoint Management formats the time display according to your computer's settings.

Status

Indicates the return code for the event.

Values: Can be one of:

- D (Denied)—Denied the event because of insufficient authorization.
- P (Permitted)—Permitted the event.
- W (Warning)—Permitted the event because Warning mode is set although the access request violates an access rule.

Event Type

Identifies the type of event this record belongs to.

Note: CA Access Control Endpoint Management refers to this field simply as *Event*.

Service

Identifies the name of the service that the connection used.

Details

Indicates at which stage CA Access Control decided what action to take for this event.

Note: The audit record you see in a non-detailed seaudit output displays a number in this field. This number is known as the authorization stage code. In a detailed output or in CA Access Control Endpoint Management, the audit record displays the message associated with the authorization stage code. For a complete list of stage codes, run seaudit -t.

Reason

Indicates the reason that CA Access Control wrote an audit record.

Note: This field does not display in a detailed seaudit output or in CA Access Control Endpoint Management. The audit record you see in a non-detailed seaudit output displays a number in this field. This number is known as the reason code. For a complete list of reason codes, run seaudit -t.

Host name

Identifies the name of the host the network traffic originated from.

Program

(UNIX only) Identifies the name of the program the accessor is attempting to run.

Example: Inbound Network Connection Event Message

The following audit record was taken from a detailed seaudit output.

```
17 Nov 2008 12:22:04 D HOST         telnet              173  3 computer.org.com
/usr/sbin/inetd
Event type: Inbound network connection
Status: Denied
Host name: computer.org.com
Service: telnet
Program: /usr/sbin/inetd/
Date: 17 Nov 2008
Time: 12:22
Details: HOST entry day & time restrictions
Audit flags: AC database user
```

This audit record indicates that on November 17th 2008, an accessor attempting to access the host computer.org.com using the telnet service to run the inetd program was denied due to day and time restrictions imposed on the protected host (authorization stage code 173—HOST entry day & time restrictions). CA Access Control logged this event because the resource's audit mode specifies that this event should be logged (reason code 3—Resource audit mode required logging).

More information:

Authorization Stage Codes for Inbound Network Connection Events (see page 595)
Reason Codes That Specify Why a Record Was Created (see page 614)

Outbound Network Connection Event

Outbound network connection events indicate outbound traffic to the protected host. Outbound network events are audited in two forms (according to the class activation in the local database). Both audit event types contain identical information but in different view. For example, one audit event contains HOST as the class name while the other event displays TCP as the class name.

Audit records in this event have the following format:

Date Time Status Class Service UserName Details Reason Host Program Terminal AuditFlags

Date

Identifies the date the event occurred.

Format: DD MMM YYYY

Note: CA Access Control Endpoint Management formats the date display according to your computer's settings.

Time

Identifies the time the event occurred.

Format: HH:MM:SS

Note: CA Access Control Endpoint Management formats the time display according to your computer's settings.

Status

Indicates the return code for the event.

Values: Can be one of:

- D (Denied)—Denied the event because of insufficient authorization.
- P (Permitted)—Permitted the event.
- W (Warning)—Permitted the event because Warning mode is set although the access request violates an access rule.

Class

Identifies the name of the class.

Service

Identifies the name of the service that the connection used.

User Name

Identifies the name of the accessor that performed the action that triggered this event.

Details

Indicates at which stage CA Access Control decided what action to take for this event.

Note: The audit record you see in a non-detailed seaudit output displays a number in this field. This number is known as the authorization stage code. In a detailed output or in CA Access Control Endpoint Management, the audit record displays the message associated with the authorization stage code. For a complete list of stage codes, run seaudit -t.

Reason

Indicates the reason that CA Access Control wrote an audit record.

Note: This field does not display in a detailed seaudit output or in CA Access Control Endpoint Management. The audit record you see in a non-detailed seaudit output displays a number in this field. This number is known as the reason code. For a complete list of reason codes, run seaudit -t.

Host name

Identifies the name of the target host.

Program

Identifies the name of the program that triggered the event.

Terminal

Identifies the name of the terminal that the accessor used to connect to the host.

User Logon Session ID

Identifies the accessor's session ID.

Note: By default this field does not appear in a non-detailed seaudit output. To display this field in a non-detailed seaudit output, specify the -sessionid option in the seaudit command. The user logon session ID field is added only to events that were generated as a result of TCP or CONNECT class definitions.

Audit Flags

Indicates whether the accessor is internal (CA Access Control database user) or an enterprise user.

Note: If the accessor is an enterprise user, the audit record you see in a non-detailed seaudit output displays the string "(OS user)" in this field. Otherwise, this field remains empty.

Example: Outbound Network Connection Event Message

The following audit record was taken from a detailed seaudit output.

```
21 Jan 2009 15:37:43 D TCP      telnet    root    408  2 computer.org /usr/bin/telnet
computer.com
Event type: Outbound network connection
Status: Denied
Host name: computer.org
Service: telnet
Program: /usr/bin/telnet
User name: Administrator
Terminal: computer.com
User name: root
Date: 21 Jan 2009
Time: 15:37:43
Details: Default access of TCP service
User Logon Session ID: 4977248c:0000012a5248
Audit flags: AC database user
```

This audit record indicates that on January 21st, 2009, the administrator opened an outgoing connection from the terminal computer.org to the computer named computer.com via the telnet service. CA Access Control denied this operation because of the defaccess property of the TCP record. (authorization stage code 408—Default of TCP service).CA Access Control logged this event because the AUDIT_MODE property for the accessor matches the record's result. (reason code 2—User audit mode requires logging).

More information:

Authorization Stage Codes for Outbound Network Connection Events (see page 599)
Reason Codes That Specify Why a Record Was Created (see page 614)

Security Database Administration Event

Security database administration events describe actions performed by a CA Access Control administrator or a sub-administrator with appropriate privileges that were intercepted by CA Access Control.

Audit records in the event have the following format:

Date Time Status Event Class Admin Details Reason Object Terminal Command AuditFlags

Date

Identifies the date the event occurred.

Format: DD MMM YYYY

Note: CA Access Control Endpoint Management formats the date display according to your computer's settings.

Time

Identifies the time the event occurred.

Format: HH:MM:SS

Note: CA Access Control Endpoint Management formats the time display according to your computer's settings.

Status

Indicates the return code for the event.

Values: Can be one of:

- D (Denied)—Denied the event because of insufficient authorization.
- S (Success)—Permitted the event.
- F (Failed)—Failed the event.

Event Type

Identifies the type of event this record belongs to.

Note: CA Access Control Endpoint Management refers to this field simply as *Event*.

Class

Identifies the class that the resource being administered belongs to.

Administrator

Identifies the name of the administrative user that executed the selang command.

Details

Indicates at which stage CA Access Control decided what action to take for this event.

Note: The audit record you see in a non-detailed seaudit output displays a number in this field. This number is known as the authorization stage code. In a detailed output or in CA Access Control Endpoint Management, the audit record displays the message associated with the authorization stage code. For a complete list of stage codes, run seaudit -t.

Reason

Indicates the reason that CA Access Control wrote an audit record.

Note: This field does not display in a detailed seaudit output or in CA Access Control Endpoint Management. The audit record you see in a non-detailed seaudit output displays a number in this field. This number is known as the reason code. For a complete list of reason codes, run seaudit -t.

Object

Identifies the name of the resource that is being administrated.

Terminal

Identifies the name of the terminal that the accessor used to connect to the host.

Note: If the command originated from a parent policy model, this field displays the fully qualified PMD name.

Command

Displays the selang command that the user executed.

Audit Flags

Indicates whether the accessor is internal (CA Access Control database user) or an enterprise user.

Note: If the accessor is an enterprise user, the audit record you see in a non-detailed seaudit output displays the string "(OS user)" in this field. Otherwise, this field remains empty.

Command type

Identifies the type of the database administration command that this event describes.

Values can be one of:

- **Add user**—For newusr command
- **Add group**—For newgrp command
- **Add resource**—For newres or newfile commands
- **Modify user**—For chusr command

- **Modify group**—For chgrp command
- **Modify group membership**—For join command
- **Modify resource**—For chres command
- **Modify resource access**—For authorize command
- **Remove user**—For rmusr command
- **Remove group**—For rmgrp command
- **Remove resource**—For rmres or rmfile commands
- **Set options**—For setoptions command
- **Add/Modify user**—For editusr command
- **Add/Modify group**—For editgrp command
- **Add/Modify resource**—For editres or editfile commands
- **Administrative command**—For other commands

Example: Security Database Administration Event Message

The following audit record was taken from a detailed seaudit output.

```
05 Nov 2008 15:45:12 S UPDATE       FILE       DOMAIN_NAME\computer 305  0 dfdok
computer.com cr file dfdok defacc(r)
Event type: Security database administration
Command type: Modify resource
Status: Successful
Administrator: DOMAIN_NAME\computer
Class: FILE
Object: dfdok
Terminal: computer.com
Date: 05 Nov 2008
Time: 15:45
Details: Command successful for ADMIN user.
Command: cr file dfdok defacc(r)
Audit flags: AC database user
```

This audit record indicates that on November 5th 2008, CA Access Control denied access from an administrator attempting to update a file by executing the command cr file dfdok defacc(r) on the protected host logging from the terminal computer.com (authorization stage code 305—Command allowed for ADMIN user).

More information:

Authorization Stage Codes for Security Database Administration Events (see page 602)
Reason Codes That Specify Why a Record Was Created (see page 614)

Startup Event

CA Access Control startup events describe the startup sequence of CA Access Control services (Windows) or daemons (UNIX).

Audit records in the event have the following format:

Date Time M *Event Service*

Date

Identifies the date the event occurred.

Format: DD MMM YYYY

Note: CA Access Control Endpoint Management formats the date display according to your computer's settings.

Time

Identifies the time the event occurred.

Format: HH:MM:SS

Note: CA Access Control Endpoint Management formats the time display according to your computer's settings.

Event Type

Identifies the type of event this record belongs to.

Note: CA Access Control Endpoint Management refers to this field simply as *Event*.

Service

seosd - the main CA Access Control daemon or service. The seosd daemon or service controls the start up and shutdown sequences of CA Access Control.

Example: Daemon Start Event Message (UNIX)

The following audit record was taken from a detailed seaudit output.

```
02 Nov 2008 15:41:06 M START                                    seoswd
Event type: Daemon start
Daemon: seoswd
Date: 02 Nov 2008
```

```
Time: 15:41
Audit flags: AC database user
```

This audit record indicates that on November 2nd 2008 the seoswd Watchdog started.

Example: Engine Service Start Event Message (Windows)

The following audit record was taken from a detailed seaudit output.

```
02 Nov 2008 15:34:48 M START                                    seosd
Event type: Engine service start
Engine service: seosd
Date: 02 Nov 2008
Time: 15:34
Audit flags: AC database user
```

This audit record indicates that on November 2nd 2008, the seosd service engine, responsible for starting up CA Access Control, started.

Shutdown Event

CA Access Control shutdown events describe shutdown processes performed by an administrator or sub-administrator user with privileges to shutdown the system.

Audit records in this event have the following format:

Date Time M *Event UserName SessionID Details Service AuditFlags*

Date

Identifies the date the event occurred.

Format: DD MMM YYYY

Note: CA Access Control Endpoint Management formats the date display according to your computer's settings.

Time

Identifies the time the event occurred.

Format: HH:MM:SS

Note: CA Access Control Endpoint Management formats the time display according to your computer's settings.

Event Type

Identifies the type of event this record belongs to.

Note: CA Access Control Endpoint Management refers to this field simply as *Event*.

User Name

Identifies the name of the accessor that performed the action that triggered this event.

User Logon Session ID

Identifies the accessor's session ID.

Note: By default this field does not appear in a non-detailed seaudit output. To display this field in a non-detailed seaudit output, specify the -sessionid option in the seaudit command.

Details

Indicates at which stage CA Access Control decided what action to take for this event.

Note: The audit record you see in a non-detailed seaudit output displays a number in this field. This number is known as the authorization stage code. In a detailed output or in CA Access Control Endpoint Management, the audit record displays the message associated with the authorization stage code. For a complete list of stage codes, run seaudit -t.

Daemon (UNIX) / Engine service (Windows)

Identifies the name of the CA Access Control daemon (UNIX) or service (Windows) that was shut down.

Value: seosd (the CA Access Control Engine).

Audit Flags

Indicates whether the accessor is internal (CA Access Control database user) or an enterprise user.

Note: If the accessor is an enterprise user, the audit record you see in a non-detailed seaudit output displays the string "(OS user)" in this field. Otherwise, this field remains empty.

Example: Shutdown Event Message on UNIX

The following audit record was taken from a detailed seaudit output.

```
24 Sep 2008 15:40:46 M SHUTDOWN     root    452 seosd
Event type: Daemon shutdown
User name: root
Daemon: seosd
Date: 24 Sep 2008
Time: 15:40:46
Details: User is ADMIN or SPECIAL
User Logon Session ID: 48da26ce:00000142
Audit flags: CA Access Control database user
```

This audit record indicates that on September 24rd 2008, the user root who was attempting to shutdown CA Access Control was permitted to do so because the user has the ADMIN attribute (authorization stage code 452—User is ADMIN or SPECIAL).

Example: Shutdown Event Message on Windows

The following audit record was taken from a detailed seaudit output.

```
23 Dec 2008 12:56:20 D SHUTDOWN     tst002                  460 seosd
Event type: Engine service shutdown
User name: tst002
Engine service: seosd
Date: 10 Feb 2009
Time: 12:56
Details: User is not allowed to shutdown CA Access Control

User Logon Session ID: 00000000:04c240d5
Audit flags: AC database user
```

This audit record indicates that on December 23rd 2008, the CA Access Control shut down was denied because the user tst002 is not allowed to shutdown CA Access Control (authorization stage code 460—User is not allowed to shutdown CA Access Control).

More information:

Authorization Stage Codes for Shutdown Events (see page 608)

Password Verification Event

Password verification event type messages indicate that a user failed to change his account's password.

Audit records in this event have the following format:

Date Time Status Event UserName Details Reason AuditFlags

Date

Identifies the date the event occurred.

Format: DD MMM YYYY

Note: CA Access Control Endpoint Management formats the date display according to your computer's settings.

Time

Identifies the time the event occurred.

Format: HH:MM:SS

Note: CA Access Control Endpoint Management formats the time display according to your computer's settings.

Status

Indicates the return code for the event.

Value: F (Failed)—Failed to change the account password.

Event Type

Identifies the type of event this record belongs to.

Note: CA Access Control Endpoint Management refers to this field simply as *Event*.

User Name

Identifies the name of the user to which the password attempt was applied.

Details

Indicates why the password change attempt failed.

Note: The audit record you see in a non-detailed seaudit output displays a number in this field. This number is known as the password quality code. In a detailed output or in CA Access Control Endpoint Management, the audit record displays the message associated with the password quality code. For a complete list of password quality codes, run seaudit -t.

Reason

Indicates the reason that CA Access Control wrote an audit record.

Note: This field does not display in a detailed seaudit output or in CA Access Control Endpoint Management. The audit record you see in a non-detailed seaudit output displays a number in this field. This number is known as the reason code. For a complete list of reason codes, run seaudit -t.

Audit Flags

Indicates whether the accessor is internal (CA Access Control database user) or an enterprise user.

Note: If the accessor is an enterprise user, the audit record you see in a non-detailed seaudit output displays the string "(OS user)" in this field. Otherwise, this field remains empty.

Example: Password Verification Event Message

The following audit record was taken from a detailed seaudit output.

```
02 Dec 2008 10:23:47 F PASSWORD     test1         1 10
Event type: Password verification
Status: Failed
User name: test1
Details: Password too short
Audit flags: AC database user
```

This audit record indicates that on December 2nd 2008, the user attempting to change his account password was denied because the password did not meet the minimum required number of characters, as defined by the password policy (authorization stage code 1—Password too short). CA Access Control logged this event message according to an explicit request (reason code 10—An explicit request to log the operation was received).

More information:

Authorization Stage Codes for Password Verification Events (see page 609)
Reason Codes That Specify Why a Record Was Created (see page 614)

Trace Message On a User

Trace messages on user events describe an attempt to open, run, or use a protected resource.

Audit records in this event have the following format for Windows:

Date Time Status Event UserName SessionID RealUID RealUsername Class Resource Details Trace AuditFlags

Audit records in this event have the following format for UNIX:

Date Time Status Event UserName SessionID EffectiveUsername RealUsername Class Resource Details Trace AuditFlags

Date

Identifies the date the event occurred.

Format: DD MMM YYYY

Note: CA Access Control Endpoint Management formats the date display according to your computer's settings.

Time

Identifies the time the event occurred.

Format: HH:MM:SS

Note: CA Access Control Endpoint Management formats the time display according to your computer's settings.

Status

Indicates the return code for the event.

Values: Can be one of:

- D (Denied)—Denied the event because of insufficient authorization.
- P (Permitted)—Permitted the event.
- W (Warning)—Permitted the event because Warning mode is set although the access request violates an access rule.

Note: In a detailed seaudit output this field displays the trace information.

Event Type

Identifies the type of event this record belongs to.

Note: CA Access Control Endpoint Management refers to this field simply as *Event*.

User Name

Identifies the name of the accessor that performed the action that triggered this event.

User Logon Session ID

Identifies the accessor's session ID.

Real User ID

Identifies the user ID of the user who invoked the process.

Note: (UNIX) This field does not appear in non detailed seaudit output.

Real user name

Identifies the name of the user performing the traced action.

Effective user ID

(UNIX only) Indicates the ID of the native OS effective user ID.

Note: This field does not appear in non detailed seaudit output.

Effective User Name

Identifies the name of the native OS effective user that triggered this event. This is different from the user name if the user substitutes (surrogates) to a different user or runs a setuid program.

Class

Identifies the class that the resource being accessed belongs to.

Resource

Identifies the name of the actual resource that is being accessed or updated.

Details

Indicates at which stage CA Access Control decided what action to take for this event.

Note: The audit record you see in a non-detailed seaudit output displays a number in this field. This number is known as the authorization stage code. In a detailed output or in CA Access Control Endpoint Management, the audit record displays the message associated with the authorization stage code. For a complete list of stage codes, run seaudit -t.

Trace information

Displays the trace detail information including the class, resource, and action that was performed on that resource or the result of that action.

Audit Flags

Indicates whether the accessor is internal (CA Access Control database user) or an enterprise user.

Note: If the accessor is an enterprise user, the audit record you see in a non-detailed seaudit output displays the string "(OS user)" in this field. Otherwise, this field remains empty.

Example: Trace Message On a User Event Message on UNIX

The following audit record was taken from a detailed seaudit output.

```
03 Nov 2008 10:38:47 P TRACE        root         490daddd:00000140 john         root
FILE       /home/jon/file.txt   55  FILE    > Result: 'P' [stage=55 gstag=55 ACEEH=8
rv=0(/home/john/file.txt
Event type: Trace message on a user
Date: 03 Nov 2008
Time: 10:38
Details: Resource ACL check
Trace information: FILE    > Result: 'P' [stage=55 gstag=55 ACEEH=8
rv=0(/home/john/file.txt
Class: FILE
Resource: /home/admin/file.txt
User name: root
Real user ID: 108
Real user name: john
Effective user ID: 108
Effective user name: root
User Logon Session ID: 490daddd:00000140
Audit flags: AC database user
```

This audit record indicates that on November 3rd 2008, a trace message was logged due to an administrator attempt to access a resource belonging to a FILE class. The administrator was permitted to access according to the ACL of the accessed resource (authorization stage code 55—Resource ACL check).

Example: Trace Message On a User Event Message on Windows

The following audit record was taken from a detailed seaudit output.

```
10 Nov 2008 10:14:53 P TRACE     MACHINE\Administrator 00000000:172ef9ef MACHINE\john
MACHINE\john WINSERVICE   _default     1059  WINSERVICE >
(C:\WINDOWS\system32\services.exe) Result: 'P' [stage=1059 gstag=1059 ACEEH=6
rv=0x0 (WebClient)]                    Why?  Default record universal access check
Event type: Trace message on a user
Date: 10 Nov 2008
Time: 10:14
Details: Default record universal access check
Trace information: WINSERVICE > (C:\WINDOWS\system32\services.exe) Result: 'P'
[stage=1059 gstag=1059 ACEEH=6    rv=0x0 (WebClient)]                 Why?  Default
record universal access check
Class: WINSERVICE
Resource: _default
User name: MACHINE\Administrator
Real user name: MACHINE\john
```

```
User Logon Session ID: 00000000:172ef9ef
Audit flags: AC database user
```

This audit record indicates that on November 10th 2008, a trace message was triggered due to an administrator attempting to access the resource _default belonging to the WINSERVICE class. The administrator was permitted access because of a record universal access check (authorization stage code 1059—Default record universal access check).

More information:

Authorization Stage Codes for Trace Message On a User (see page 613)
Reason Codes That Specify Why a Record Was Created (see page 614)

Authorization Stage Codes for Log In and Log Out Events

Authorization stage codes for log in and log out events describe at which stage CA Access Control decided what action to take for the log in or log out event.

More information:

Login Event (see page 546)
Logout Event (see page 548)
Login Account Enabled Event (see page 551)
Login Account Disabled Event (see page 553)
Password Attempt Event (see page 555)

2—Fetching user object

Indicates that a login attempt failed because CA Access Control could not load user information, such as user mode, terminal, or login program. CA Access Control may write this message to the audit log if the database is corrupt or CA Access Control did not start correctly.

3—Terminal checking for login terminal source

Indicates that CA Access Control permitted or denied login according to the TERMINAL class rules.

5—User suspend checking

Indicates that CA Access Control denied login, because the user account is suspended.

6—User expiration checking

Indicates that CA Access Control denied login, because the user account is expired, as defined in the user's profile.

7—User day-time checkings

Indicates that CA Access Control denied login, because the user attempted to log in at a time outside the permitted day and time for the CA Access Control database.

8—Password validity checkings

Valid on UNIX

Indicates that CA Access Control checked a user's password to ensure it conformed to the password rules. CA Access Control may write this message to the audit log when a login attempt failed because a user's password did not conform to the CA Access Control database password rules.

9—User grace login checkings

Indicates that CA Access Control denied login, because the user account has exhausted its grace login attempts.

10—Password expired with no more grace logins

Indicates that CA Access Control denied login, because the password is expired. The user did not change their password within the password interval limit and no grace count after password expiration is configured, neither in the user's profile group's definition nor in the CA Access Control global definitions.

11—Building the user ACEE

Indicates that CA Access Control successfully generated the ACEE for the user.

12—User inactivity days check

Indicates that CA Access Control denied login, because the user was inactive for a period that exceeded the permitted inactive interval. The permitted inactive interval is defined in the user's profile or global CA Access Control settings.

13—Too many logins for user

Indicates that CA Access Control denied login, because the user has exceeded the maximum allowed number of simultaneous logins from different terminals. The maximum allowed number of simultaneous logins is defined in the 'Maxlogins' properly value in the user's profile or global CA Access Control settings.

14—Active HOLIDAY check

Indicates that CA Access Control denied login, because the user attempted to log in during the restricted holiday dates. The restricted holiday dates are defined in the HOLIDAY class.

15—Login Application (LOGINAPPL) check

Valid on UNIX

Indicates that CA Access Control denied login, because of the LOGINAPPL class rules.

16—User Groups day-time checking

Indicates that CA Access Control denied login, because the user attempted to log in at a time outside the permitted day and time for the user or for one of the user's group.

17—Attempt rejected by the native environment

Valid on UNIX

Indicates that the login attempt failed due to the native environment settings. Logged by CA Access Control PAM module.

18—User without domain restriction

Valid on Windows

Indicates that CA Access Control denied login, because the user did not provide a domain name.

19—No reason to deny – allow login

Indicates that CA Access Control permitted login, because the login attempt passed all check stages, providing that the login authorization has a TERMINAL object assigned to.

Note: Viewing this event stage message may indicate that the login authorization was triggered by CA Access Control authorization API that does not have the terminal name specified.

20—'Logical' user check

Indicates that CA Access Control denied login, because CA Access Control does not permit 'logical' users (users with the *logical* property set) to log in.

49—Logout detected after last process terminated

Valid on UNIX

Indicates that CA Access Control detected a user logout event occurring after the last process terminated.

Authorization Stage Codes for Resource Access Events

Authorization stage codes for resource access events describe at which stage CA Access Control decided to take action for the resource access event.

More information:

Resource Access Event (see page 557)

50—Security LABEL check of resource

Indicates that CA Access Control denied access to the resource, because *one* of the following is true for the user who tried to access the resource:

- The resource security label has a higher security level than the user security label
- The user does not have a security label

51—Security LEVEL check of resource

Indicates that CA Access Control denied access to the resource, because *one* of the following is true for the user who tried to access the resource:

- The resource has a higher security level than the user
- The user does not have a security level

52—Category check of resource

Indicates that CA Access Control denied access to the resource, because the resource is assigned a security category that is not assigned to the user.

53—Resource DAYTIME check

Indicates that CA Access Control denied access to the resource, because the user attempted access at a time outside the permitted day and time for the resource.

54—OWNER check of resource

Indicates that CA Access Control permitted access to a resource, because the accessing user owns the resource.

55—Resource ACL check

Indicates that CA Access Control permitted or denied access to the resource, because the resource ACL lists the user.

56—In resource group ACL check

Indicates that CA Access Control permitted or denied access to the resource, because the resource group ACL lists list the user.

57—User group in resource ACL

Indicates that CA Access Control permitted or denied access to the resource because the user group ACL list at least one of the resource.

58—User group in resource group ACL

Indicates that CA Access Control permitted or denied access to the resource, because the resource group ACL lists at least one of the user group.

59—Resource UACC check

Indicates that CA Access Control permitted access to the resource, because of the resource's default settings.

61—User is OPERATOR on resource

Indicates that CA Access Control permitted access to the resource, because the user has the OPERATOR attribute. The OPERATOR attribute lets users bypass authorization procedures for read and chdir access for FILE resources.

Note: On UNIX, CA Access Control writes this message to the trace file only, and does not write the message to the audit log file.

62—UACC check for Class of unprotected resource

Indicates that CA Access Control permitted or denied access to a resource that does not have a record in the CA Access Control database, based on the defaccess value in the resource class.

63—Program Conditional Access

Indicates that CA Access Control permitted or denied access to the resource, because the resource PACL lists the program and the user or one of the user's groups.

64—User '*' in resource ACL

Indicates that CA Access Control permitted or denied access to the resource, because the resource ACL contains an asterisk (*).

Note: An asterisk specifies all defined users.

65—User is AUDITOR on resource

Indicates that CA Access Control permitted access to the audit file, because the user has the AUDITOR attribute. The AUDITOR attribute lets users bypass authorization procedures for read and chdir access requests.

Note: CA Access Control writes this message to the trace file only, and does not write the message to the audit log file.

69—No step that allowed access

Indicates that CA Access Control denied access to the resource because it could not find a rule that let the user access the resource.

70—OWNER check of resource's group

Indicates that CA Access Control permitted access to the resource because the user attempting to access the resource is the owner of one of the resource's groups.

75—User '*' in resource group ACL

Indicates that CA Access Control permitted or denied access to the resource because the resource group ACL contains an asterisk (*).

Note: An asterisk specifies all defined users.

76—Resource denied ACL check

Indicates that CA Access Control denied access to the resource, because the resource NACL lists the user.

77—In resource group denied ACL check

Indicates that CA Access Control denied access to the resource, because the resource group NACL lists the user.

78—User group in resource denied ACL

Indicates that CA Access Control denied access to the resource, because the resource NACL lists at least one of the the user group.

79—User group in resource group denied ACL

Indicates that CA Access Control denied access to the resource, because the resource group NACL lists at least one of the user groups.

80—User '*' in resource denied ACL

Indicate that CA Access Control denied access to the resource, because the resource NACL contains an asterisk (*).

Note: An asterisk specifies all defined users.

81—User '*' in resource group denied ACL

Indicates that CA Access Control denied access to the resource, because the resource group NACL contains an asterisk (*).

Note: An asterisk specifies all defined users.

82—Group of resource DAYTIME check

Indicates that CA Access Control denied access to the resource, because the user attempted to access the resource at a time outside the permitted day and time for the resource group.

86—Resource calendar ACL check for user

Indicates that CA Access Control permitted or denied access to the resource, because the user attempted to access the resource at a time permitted or denied by the resource CALACL.

87—Resource group calendar ACL check for user

Indicates that CA Access Control permitted or denied access to the resource, because the user attempted to access the resource at a time permitted or denied by the resource group CALACL.

88—Resource calendar ACL check for user groups

Indicates that CA Access Control permitted or denied access to the resource, because the user attempted to access the resource at a time permitted or denied because the user is a member of one of the groups that are listed in the resource CALACL.

89—Resource group calendar ACL check for user groups

Indicates that CA Access Control permitted or denied access to the resource, because the user group attempted to access the resource at a time permitted or denied because the user is a member in one of the group's that are listed in the resource CALACL.

90—User * in resource calendar ACL

Indicates that CA Access Control permitted or denied access to the resource, because the resource CALACL contains an asterisk (*).

Note: An asterisk specifies all defined users.

91—User * in resource groups calendar ACL

Indicates that CA Access Control permitted or denied access to the resource, because the resource group CALACL contains an asterisk (*).

Note: An asterisk specifies all defined users.

92—Attempt to rename the path of a protected resource

Valid on Windows

Indicates that CA Access Control denied a request to rename a protected file or registry entry.

200—Class checks not active

Indicates that CA Access Control permitted access to a resource, because the resource class is inactive.

Note: When a resource class is inactive, the setoptions list command displays the class activity as 'No'.

201—Loading the user information

Indicates that CA Access Control could not authorize a request, because it failed to retrieve a user's information.

202—Resource in WARNING mode

Indicates that CA Access Control permitted access to a resource, because the resource is in Warning Mode.

203—Access for the resource is MAXIMUM_ALLOWED

Valid on Windows

When permitted, indicates that CA Access Control assigned maximum access rights to the registry handle.

When denied, indicates that CA Access Control blocked access to the registry handle.

204—Class in WARNING mode

Indicates that CA Access Control permitted access to a resource, because the resource class is in Warning Mode.

210—Special kernel module load check

Valid on UNIX

Indicates that CA Access Control permitted or denied the loading or unloading of the kernel module, based on the KMODULE class definitions.

250—Executing an untrusted program

Indicates that CA Access Control denied an attempt to execute an untrusted program.

251—Using deniable parameter

Indicates that CA Access Control denied an attempt to execute sesudo command, because the command syntax contains parameters the SUDO record defines as prohibited.

252—Relative path specified by an _abspath user

Valid on UNIX

Indicates that CA Access Control denied an attempt to execute a program that was specified by a relative path, because the user attempting to execute the program is a member of the '_abspath' group.

253—Permitted sesudo job

Indicates that CA Access Control permitted an attempt to execute a sesudo command.

254—sesudo command failed

Valid on UNIX

Indicates that a sesudo command failed to execute on the operating system.

440—Invalid calendar was detected

Indicates that CA Access Control denied access because of an error in getting the calendar information.For example, a memory problem or calendar table corruption.

441—Calendar does not allow access

Indicates that CA Access Control denied access because the calendar object's definitions associated with the accessed resource do not allow access at this time.

1050—Default Record Security Label Check

Indicates that CA Access Control denied access to the default record, because *one* of the following is true for the user who tried to access the resource:

- The resource security label has a higher security level than the user security label
- The user does not have a security label

1051—Default Record Security Level Check

Indicates that CA Access Control denied access to the default resource, because *one* of the following is true for the user who tried to access the resource:

- The resource has a higher security level than the user
- The user does not have a security level

1052—Default Record Category Check

Indicates that CA Access Control denied access to the default resource, because the resource is assigned a security category that is not assigned to the user.

1053—Default Record Day and Time Check

Indicates that CA Access Control denied access to the default resource, because the user attempted access at a time outside the permitted day and time for the resource.

1054—Default Record OWNER Check

Indicates that CA Access Control permitted access to the default resource, because the accessing user owns the default resource.

1055—Default Record ACL Check for User

Indicates that CA Access Control permitted or denied access to the default resource, because the resource ACL lists or does not list the user.

1056—Default Record Group ACL Check For User

Indicates that CA Access Control permitted or denied access to the default resource, because the resource group ACL lists or does not list the user.

1057—Default Record ACL Check for User Groups

Indicates that CA Access Control permitted read or chdir access to the default resource.

Note: CA Access Control writes this message to the trace file only, and does not write the message to the audit log file.

1058—Default Record Group ACL Check for User Groups

Indicates that CA Access Control permitted or denied access to the default resource, because the resource group ACL lists or does not list the user group.

1059—Default Record Universal Access Check

Indicates that CA Access Control permitted access to the default resource, because of the resource's default settings.

1061—Default Record OPERATOR Attribute Check

Indicates that CA Access Control permitted access to the default resource, because the user has the OPERATOR attribute. The OPERATOR attribute lets users bypass authorization procedures for read and chdir access requests.

Note: CA Access Control writes this message to the trace file only, and does not write the message to the audit log file.

1062—Default Record Class Global Universal Access

Indicates that CA Access Control permitted or denied access to the default resource that does not have a record in the CA Access Control database, based on the defaccess value in the resource class.

1063—Default Record Program Conditional Access

Indicates that CA Access Control permitted or denied access to the default resource, because the resource PACL lists or does not list the program accessing the resource.

1064—User '*' in _default record ACL

Indicates that CA Access Control permitted or denied access to the default resource, because the resource ACL contains an asterisk (*).

Note: An asterisk specifies all defined users.

1069—No Rule Granting Access to Default Record

Indicates that CA Access Control denied access to the default resource because it could not find a rule that let the user access the resource.

1202—Default Record in WARNING Mode

Indicates that CA Access Control permitted access to the default resource, because the resource is in Warning Mode.

1250—Default Record is Set Untrusted

Indicates that CA Access Control denied an attempt to execute the default untrusted program.

Authorization Stage Codes for Untrust Message Events

Authorization stage codes for untrust message events describe at which stage CA Access Control decided what action to take for the untrust message event.

More information:

Untrust Message Event (see page 560)

0—A general error occurred during Watchdog file checking

Indicates that an error occurred while CA Access Control fetched the file information. CA Access Control may write this message to the audit log if the file is untrusted. You should check the system logs for more information.

1—Stat information of PROGRAM or SECFILE was changed

Indicates that data changed in a record in the PROGRAM or SECFILE classes. CA Access Control may write this message to the audit log if it detects an attempt to tamper with a program or file. You should check the audit events, system logs, and trace records for the program or file. If the program or file was changed by an administrator, consider re-trusting the changed program or file.

4—CRC check of PROGRAM or SECFILE changed

Indicates that the Cyclic Redundancy Check (CRC) changed of a record in the PROGRAM or SECFILE class. You should check the system logs, event log files, and trace records for the program or file.

5—Cannot stat file of PROGRAM or SECFILE

Indicates that CA Access Control failed to retrieve file information for the specified file. CA Access Control may write this message to the audit log if one of the following occurs:

- The file name or directory changed
- The file name or directory does not exist
- The access permissions of the file
- The system is out of memory

To determine the possible cause of the error, check the system log files.

7—MD5 signature of PROGRAM or SECFILE changed

Indicates that the MD5 signature changed for a record in the PROGRAM or SECFILE classes. You should check the system log files, audit messages, and trace logs for the program or file.

8—SHA1 signature of PROGRAM or SECFILE changed

Indicates that the SHA1 signature changed for a record in the PROGRAM or SECFILE classes. You should check the system log files, audit messages, and trace logs for the program or file.

Authorization Stage Codes for Inbound Network Connection Events

Authorization stage codes s for inbound network connection events describe at which stage CA Access Control decided what action to take for the incoming network connection event.

More information:

Inbound Network Connection Event (see page 563)

150—Check Class Table

Indicates that the class could not be found in the CA Access Control database. CA Access Control may write this message to the audit log if there is a problem in the CA Access Control database. To correct this problem, use the dbmgr utility to rebuild the CA Access Control database.

Important! Use the dbmgr utility only with the guidance of support personnel during problem resolution. For assistance, contact CA Support at http://ca.com/support.

More information:

dbmgr Utility (see page 28)

153—HOST entry asterisk in inetacl

Indicates that CA Access Control permitted or denied a connection from a protected host, because the host INETACL contains an asterisk (*).

Note: An asterisk specifies any sequence of zero or more characters and therefore matches all services when used in INETACL.

156—HOST entry inetacl

Indicates that CA Access Control permitted or denied a connection from the protected host, because the host INETACL lists the connection service.

157—HOST Class UACC

Indicates that CA Access Control permitted or denied a connection from the protected host, because of the default access authority value defined for the host UACC class.

159—HOST entry service range ACL

Indicates that CA Access Control permitted or denied a connection from the protected host, because the connection service is within the host INETACL range.

163—No rule granting access to service

Indicates that CA Access Control denied a connection from the host, because it did not find a rule permitting access. You should check the HOST class access rules for that host.

164—HOST group inetacl

Indicates that CA Access Control permitted or denied a connection from the protected host, because the GHOST object's INETCAL lists the connection service.

165—HOST group service range ACL

Indicates that CA Access Control permitted or denied a connection from the protected host, that is a member of the GHOST host group object, because the connection service is within the host group's INETACL range.

166—HOST group asterisk in inetacl

Indicates that CA Access Control permitted or denied a connection from a protected host that is a member of the GHOST host group object, because the host group's INETACL contains an asterisk (*).

Note: An asterisk specifies any sequence of zero or more characters and therefore matches all services when used in INETACL.

167—HOSTNET (network or IP mask/match) inetacl

Indicates that CA Access Control permitted or denied a connection from the protected host, because the HOSTNET record INETACL lists the connection service.

168—HOSTNET (network or IP mask/match) service range

Indicates that CA Access Control permitted or denied a connection from the protected host, because the connection service is within the HOSTNET record INETACL range.

169—HOSTNET (network or IP mask/match) inetacl asterisk

Indicates that CA Access Control permitted or denied a connection from a protected host, because the HOSTNET record INETACL contains an asterisk (*).

Note: An asterisk specifies any sequence of zero or more characters and therefore matches all services when used in INETACL.

170—HOSTNP (hosts name pattern) inetacl

Indicates that CA Access Control permitted or denied a connection from the protected host, because the HOSTNP record INETACL lists the connection service.

171—HOSTNP (hosts name pattern) service range

Indicates that CA Access Control permitted or denied a connection from the protected host, because the connection service is within the HOSTNP record INETACL range.

172—HOSTNP (hosts name pattern) inetacl asterisk

Indicates that CA Access Control permitted or denied a connection from a protected host, because the HOSTNP record INETACL contains an asterisk (*).

Note: An asterisk specifies any sequence of zero or more characters and therefore matches all services when used in INETACL.

173—HOST entry day & time restrictions

Indicates that CA Access Control denied access to a protected host, because the attempted access was outside the day and time restrictions in the HOST record.

174—HOST group day & time restrictions

Indicates that CA Access Control denied access to a protected host group, because of the day and time restrictions in the GHOST record.

175—HOSTNET (network or IP mask/match) day & time restrictions

Indicates that CA Access Control denied access to a protected host, because of the day and time restrictions in the HOSTNET record.

176—HOSTNP (hosts name pattern) day & time restrictions

Indicates that CA Access Control denied access to a protected host, because of the day and time restrictions in the HOSTNP record.

177—HOST_default day & time restrictions

Indicates that CA Access Control denied access to a protected host, because of the day and time restrictions in the HOST _default record.

178—HOST_default inetacl

Indicates that CA Access Control permitted or denied access to a protected host, because of the values in the HOST _default INETACL.

179—HOST_default service range

Indicates that CA Access Control permitted or denied access to a protected host, because the connection service is within the HOST _default record INETACL range.

180—HOST_default service asterisk

Indicates that CA Access Control permitted or denied access to a protected host, because the HOST _default record INETACL contains an asterisk (*).

Note: An asterisk specifies any sequence of zero or more characters and therefore matches all services when used in INETACL.

404—HOST entry in TCP service ACL

Indicates that CA Access Control permitted or denied access from a HOST, because the TCP record ACL lists the HOST.

405—GHOST entry in TCP service ACL

Indicates that CA Access Control permitted or denied access from a HOST, because the TCP record ACL lists the GHOST of which the HOST is a member.

406—HOSTNET entry in TCP service ACL

Indicates that CA Access Control permitted or denied access from a HOST, because the TCP record ACL lists the HOSTNET network of which the HOST is a part.

407—HOSTNP entry in TCP service ACL

Indicates that CA Access Control permitted or denied access from a HOST, because the TCP record ACL lists the HOSTNP set of which the HOST is a part.

Authorization Stage Codes for Outbound Network Connection Events

Authorization stage codes for outbound network connection events describe at which stage CA Access Control decided what action to take for the outbound network connection event.

More information:

Outbound Network Connection Event (see page 565)

400— _default service in class TCP

Indicates that CA Access Control permitted or denied access to a protected host, because of the _default object permissions in the TCP record for the connecting service.

401—Class UACC of TCP services

Indicates that CA Access Control permitted or denied access to a protected host, because of the value of the TCP object in the UACC class.

402—Day and time restrictions on TCP service

Indicates that CA Access Control denied access to a TCP service, because of the day and time restrictions in the TCP record.

403—ACL read stage of TCP service

Indicates that CA Access Control permitted or denied access to the TCP service, because of the ACL read property in the TCP record. CA Access Control may write this message to the audit log if the database is corrupt.

408—Default access of TCP service

Indicates that CA Access Control permitted or denied access to the TCP class service, because of the defaccess property of the TCP record.

Note: This event message also applies to incoming TCP events to indicate an inbound connection to the HOST.

409—CACL read stage of TCP service

Indicates that CA Access Control denied access to the TCP service, because of the CACL read property in the TCP record. CA Access Control may write this message to the audit log if the database is corrupt.

410—HOST entry for USER in TCP service CACL

Indicates that CA Access Control permitted or denied access to a HOSt object for a specified USER or XUSER. CA Access Control used the access rules in the CACL of the TCP service to determine whether to permit or deny access.

411—GHOST entry for USER in TCP service CACL

Indicates that CA Access Control permitted or denied access to a GHOST object for a specified USER or XUSER object. CA Access Control used the access rules in the CACL of the TCP service to determine whether to permit or deny access.

412—HOSTNET entry for USER in TCP service CACL

Indicates that CA Access Control permitted or denied access to a HOSTNET object for a specified USER or XUSER object. CA Access Control used the access rules in the CACL of the TCP service to determine whether to permit or deny access.

413—HOSTNP entry for USER in TCP service CACL

Indicates that CA Access Control permitted or denied access to a HOSTNP object for a specified USER or XUSER object. CA Access Control used the access rules in the CACL of the TCP service to determine whether to permit or deny access.

414—HOST entry for GROUP in TCP service CACL

Indicates that CA Access Control permitted or denied access to a HOST object for a specified GROUP or XGROUP object. CA Access Control used the access rules in the CACL of the TCP service to determine whether to permit or deny access.

415—GHOST entry for GROUP in TCP service CACL

Indicates that CA Access Control permitted or denied access to a GHOST object for a specified GROUP or XGROUP object. CA Access Control used the access rules in the CACL of the TCP service to determine whether to permit or deny access.

416—HOSTNET entry for GROUP in TCP service CACL

Indicates that CA Access Control permitted or denied access to a HOSTNET object for a specified GROUP or XGROUP object. CA Access Control used the access rules in the CACL of the TCP service to determine whether to permit or deny access.

417—HOSTNP entry for GROUP in TCP service CACL

Indicates that CA Access Control permitted or denied access to a HOSTNP object for a specified GROUP or XGROUP object. CA Access Control used the access rules in the CACL of the TCP service to determine whether to permit or deny access.

418—HOST entry for User '*' in TCP service CACL

Indicates that CA Access Control permitted or denied access to a HOST for a user, because the HOST record CACL contains an asterisk (*).

Note: An asterisk specifies all defined users.

419—GHOST entry for User '*' in TCP service CACL

Indicates that CA Access Control permitted or denied access to a HOST belonging to GHOST class for a user, because the GHOST record CACL contains an asterisk (*).

Note: An asterisk specifies all defined users.

420—HOSTNET entry for User '*' in TCP service

Indicates that CA Access Control permitted or denied access to a HOSTNET object for a user, because the HOSTNET record CACL contains an asterisk (*).

Note: An asterisk specifies all defined users.

421—HOSTNP entry for User '*' in TCP service CACL

Indicates that CA Access Control permitted or denied access to a HOSTNP object for a user, because the HOSTNET record CACL contains an asterisk (*).

Note: An asterisk specifies all defined users.

Authorization Stage Codes for Security Database Administration Events

Authorization stage codes for security database administration events describe at which stage CA Access Control decided what action to take for the security database administration event.

More information:

Security Database Administration Event (see page 568)

300—Undefined CA Access Control user

Indicates that CA Access Control denied access to the system, because the accessing user could not be found in the CA Access Control database. You should check the user account profile.

301—An attempt to delete last ADMIN user

Indicates CA Access Control denied a request to do one of the following:

- Delete the last ADMIN user from the CA Access Control database
- Remove the ADMIN attribute from the only user that is assigned the ADMIN attribute

302—An attempt to delete user root

Valid on UNIX

Indicates that CA Access Control denied an attempt to delete the system root account.

303—User trying to change their own password

Indicates that CA Access Control denied a user attempt to use a selang command to change their own password. On UNIX you may change your password using the sepass utility. On Windows you may change your password using native password management tools.

304—Nonauditor user trying to set audit mode

Indicates that CA Access Control denied a user attempt to change the audit mode of a record, because the user does not have the AUDITOR attribute. To let the user change the audit mode of a record, assign the user the AUDITOR attribute.

305—Command allowed for ADMIN user

Indicates that CA Access Control permitted an action, because the user requesting the action has the ADMIN attribute.

306—Showuser (myself) , Showxusr allowed

Indicates that CA Access Control permitted a user or an external user to display the properties of their own record in the CA Access Control database.

Note: This message is not written as an audit record.

307—User trying to set categories they do not have

Indicates that CA Access Control denied an attempt to assign a security category to a user, because the user attempting to assign the security category does not possess that security category themselves.

308—User trying to set a security-label they do not have

Indicates that CA Access Control denied an attempt to assign a security label to a user, because the user attempting to assign the security label does not possess that security label themselves.

309—User trying to set security-level greater than the user's own

Indicates that CA Access Control denied an attempt to assign a security level to a user, because the user has a lower security level than the security level they are attempting to assign.

310—NonADMIN user trying to set user-mode

Indicates that CA Access Control denied an attempt to set an administrative attribute, because the user attempting to set the attribute does not have the ADMIN attribute.

311—Command allowed for object owner

Indicates that CA Access Control permitted an action, because the user owns the record.

312—Native file owner can define it to CA Access Control

Valid on UNIX

Indicates that CA Access Control permitted an action, because the file owner defined the file to CA Access Control.

Note: A file owner can define a file to CA Access Control when the use_unix_file_owner token in the lang section of the seos.ini file is set to yes.

313—Command allowed for a GROUP-ADMIN user

Indicates that CA Access Control permitted a user with the GROUP-ADMIN attribute to modify a record within the group.

314—GROUP-ADMIN user can join/join- to group

Indicates that CA Access Control permitted a user with the GROUP-ADMIN attribute add or remove a user to the group.

315—GROUP-AUDITOR/ADMIN can list the group

Indicates that CA Access Control permitted a user to list the properties of a record within a group, because the user has the GROUP-ADMIN or GROUP-AUDITOR attribute for that group.

316—An auditor can list any object

Indicates that CA Access Control permitted a user with the AUDITOR attribute to display data in the database.

317—An OPERATOR can list any object

Indicates that CA Access Control permitted a user with the OPERATOR attribute to display data in the database

318—A GROUP-AUDITOR can list objects in group scope

Indicates that CA Access Control permitted a user with the GROUP-AUDITOR attribute to display data about the group in the database.

319—A GROUP-OPERATOR can list objects in group scope

Indicates that CA Access Control permitted a user with the GROUP-OPERATOR attribute to display data about the group in the database.

320—Command allowed for CLASS-ADMIN user

Indicates CA Access Control permitted the action, because the action was performed by a user listed in the ACL of the ADMIN class.

321—Command allowed for PWMANAGER/ADMIN with access

Indicates that CA Access Control permitted a user to change a password, because the user has the PWMANAGER or ADMIN attribute.

322—There is no rule allowing this operation

Indicates that CA Access Control denied an operation, because no rule that permitted the operation was found.

324—User changing their own password using sepass

Indicates that CA Access Control permitted a user to use the sepass utility or the password policy model to change their password.

326—User created 'Login Information' for themselves

Indicates that CA Access Control permitted a user to created login information for themselves.

327—Command allowed for GROUP-PWMANAGER

Indicates that CA Access Control permitted the command, because the user that executed the command has the GROUP-PWMANAGER attribute.

329—A PWMANAGER enabled a user

Indicates that CA Access Control permitted a user to enable (re-activate) another user, because the user that enabled the other user has the PWMANAGER attribute.

330—Command allowed for DOMAIN change

Valid on Windows

Indicates that CA Access Control permitted the user to change the DOMAIN class, for example, adding new computers to the domain.

331—Command allowed for PWMANAGER

Indicates that CA Access Control permitted the command to execute, because the user that executed the command has the PWMANAGER attribute.

332—Changing native flags allowed for PWMANAGER

Valid on Windows

Indicates that CA Access Control permitted the user to modify the account flags assigned to a user account, because the user has the PWMANAGER attribute.

333—Changing 'must change password next logon' attribute is allowed for PWMANAGER

Valid for Windows

Indicates that CA Access Control permitted the user to modify the 'must change password next logon' attribute for a user account, because the user has the PWMANAGER attribute.

334—Command allowed for GROUP-PWMANAGER

Indicates that CA Access Control permitted the command, because the user that executed the command has the GROUP-PWMANAGER attribute.

335—Editing 'Login Information' is allowed for PWMANAGER

Indicates that CA Access Control permitted the user to edit the 'Login Information' attribute for a user account, because the user has the PWMANAGER attribute.

336—Command allowed for auditor user

Indicates that CA Access Control permitted a user to execute a command, because the user has the AUDITOR attribute.

337—Failed to reconcile command with database information

Indicates that the CA Access Control did not execute a command, because the objects embedded in the command do not exist in the CA Access Control database. You should check the command syntax before you re-execute the command.

338—Creating a command from an implicit request

Indicates that CA Access Control created a command that originated from an implicit request.

339—SEOS_syscall module unload readiness check

Valid on UNIX

Indicates that an accessor is executing the 'secons –scl' command to check if there are processes running in the intercepted syscalls. CA Access Control dos not permit unloading the SEOS_syscall module.

Authorization Stage Codes for Shutdown Events

Authorization stage codes for shutdown events describe at which stage CA Access Control decided what action to take for the shutdown event.

More information:

Shutdown Event (see page 572)

451—User is an OPERATOR

Indicates that CA Access Control permitted the shutdown request, because the user that executed the shutdown sequence has the OPERATOR attribute.

452—User is ADMIN or SPECIAL

Indicates that CA Access Control permitted the shutdown request, because the user executing the shutdown sequence has the ADMIN attribute assigned to him.

453— _seagent is allowed to shutdown CA Access Control

Valid on UNIX

Indicates that CA Access Control permitted the shutdown request, because _seaqent is permitted to shut down CA Access Control.

460—User is not allowed to shutdown CA Access Control

Indicates that CA Access Control denied the shutdown request, because the requesting user is not permitted to shut down CA Access Control.

600—Attempting to Terminate CA Access Control

Indicates that CA Access Control denied the shutdown request, because the user attempted to terminate CA Access Control by executing the kill command.

Authorization Stage Codes for Password Verification Events

Authorization stage codes for password verification events describe at which stage CA Access Control decided what action to take for the password verification event.

More information:

Password Verification Event (see page 575)

0—Password quality verified

Indicates that the user successfully changed their password, and that the new password meets all of the password quality rules.

1—Password too short

Indicates that the password change failed, because the length of the new password does not comply with the password policy for minimum password length.

2—Password contains user name

Indicates that the password change failed, because the new password contains the user's user name.

3—Too few lowercase letters in password

Indicates that the password change failed, because the new password does not contain enough lower case letters according to the minimum defined in the password policy.

4—Too few capital letters in password

Indicates that the password change failed, because the new password does not contain enough capital letters according to the minimum defined in the password policy.

5—Too few numeric characters in password

Indicates that the password change failed, because the new password does not contain enough numeric characters according to the minimum defined in the password policy.

6—Too few other characters in password

Indicates that the password change failed, because the new password does not contain enough other characters according to the minimum defined in the password policy.

7—Too many repetitions of same char in password

Indicates that the password change failed, because the new password contains too many repeating characters according to the maximum defined in the password policy.

8—Same as current password

Indicates that the password change failed, because the new password is the same as the current password. You should select a password that you have not used before.

9—Password previously used. Select a different password

Indicates that the password change failed, because the new password was previously used. You should select a password that you have not used before.

10—Too few alphabetic characters in password

Indicates that the password change failed, because the new password does not contain enough alphabetic characters according to the minimum defined in the password policy.

11—Too few alphanumeric characters in password

Indicates that the password change failed, because the new password does not contain enough alphanumeric characters according to the minimum defined in the password policy.

12—Password was changed recently, cannot be changed again at this time

Indicates that the password change failed, because the password was recently changed and cannot be changed at this time. You should change the password only after the minimal password age period has passed according to the minimum defined in the password policy.

13—Password is contained by a previous password or vice versa

Indicates that the password change failed, because the new password contains a previous password or is part of a previous password. You should ensure that the new password does not contain a previous password, and is not part of a previous password.

14—Password contains previous password pattern

Indicates that the password change failed, because the new password contains pattern from the previous password according to the sub_str_len defined in the password policy.

16—Password too long

Indicates that the password change failed, because the new password is too long according to the maximum defined in the password policy.

20—Passwords do not match

Indicates that the password change failed, because the new password does not match the password entered in the confirm password field.

21—Cannot include predefined prohibited characters

Indicates that the password change failed, because the new password contains prohibited characters according to the password policy.

22—Password previously used

Indicates that CA Access Control denies access because the password that you entered was used before. Make sure that the new password you use conforms with the password policy rules.

23—Password is contained by a previous password or vice versa

Indicates that the password change attempt failed because the password used is contained by a previous password or that the previous password is contained in the new password. You should select a new password that does not contain a previously used password.

24—Password is in dictionary file

Indicates that the password change failed, because the new password is defined in the DICTIONARY class or DICTIONARY file. You should select a password that is not defined in the DICTIONARY class or in the DICTIONARY file.

100—Bad arguments

Indicates that the password change failed, because invalid data was sent to the authorization engine.

CA Access Control may write this message to the audit log when one of the following occurs:

- A memory problem
- A mismatch between versions of CA Access Control various modules to a recent upgrade of CA Access Control

Verify that there are no mixed CA Access Control environments and that the client and server use the same version of CA Access Control. For assistance, contact CA Support at http://ca.com/support.

Authorization Stage Codes for Trace Message On a User

Authorization stage codes for trace events on a user describe at which stage CA Access Control decided what action to take for the user activity event.

994—Informational Message

Indicates that a user accessed the trace audit records.

Note: This is an informative message only, viewed by running the seaudit –tr command.

995—Unauthorized Access to Internal Resource

Indicates that an accessor attempted an unauthorized access to an internally protected FILE resource. For example, seos.audit records.

996—Authorized Access to Internal Resource

Indicates that CA Access Control permitted access to the resource by an internal bypass. For example: reading /etc/passwd.

997—User Can Execute a setuid\setgid Directory

UNIX only

Indicates that CA Access Control bypassed an event because an accessor attempted to execute a directory marked with a setuid\setgid flag bit. This stage is part of a TRACE record message.

998—Authorization is Configured as 'Audit Mode Only'

Windows only

Indicates that CA Access Control is set to work in 'Audit Mode Only'.

999—Resource not Protected (Check if Rules Exists)

Indicates that CA Access Control permits access to an unprotected resource.

Reason Codes That Specify Why a Record Was Created

Reason codes that specify why a record was created describe at which stage CA Access Control decided what audit record to create for the event.

0—No specific request to log the operation

Indicates that CA Access Control logged this operation by default, because no specific request to log in the operation exists.

2—User audit mode requires logging

Indicates that CA Access Control logged the operation because the audit property of the accessor or its profile matches the record's result. For example, an action performed by a user with the FAILURE value set for the AUDIT_MODE property is logged only when the user fails to access a protected resource.

3—Resource audit mode required logging

Indicates that CA Access Control logged the operation because the RAUDIT property of the resource matches the record's result.

4—Resource in WARNING mode

Indicates that CA Access Control logged this operation because a WARNING property was set to the resource or to the resource's class.

5—CA Access Control serevu utility requested auditing

Valid on UNIX

Indicates that CA Access Control logged this operation because the serevu utility requested the audit record, for example, when a user attempt to log in fails.

7—Outbound connection record

Valid on UNIX

Indicates that CA Access Control logged this operation because a successful outbound connection occurred.

8—CA Access Control pam support UNIX failed login

Valid on UNIX

Indicates that CA Access Control logged this operation because the CA Access Control PAM module requested the audit, for example, in an event of a failed password login attempt.

9—Daytime restrictions check of CALENDAR class

Indicates that CA Access Control logged this message because of a daytime restrictions check of a CALENDAR class required logging an audit record.

10—A specific request to log operation

Indicates that CA Access Control logged this operation because of a specific request to log the operation, for example, attempting to kill the CA Access Control daemons.

11—CA Access Control secons utility requested auditing

Valid on UNIX

Indicates that CA Access Control logged this operation because the Syscall monitor option is sued (secons-scl).

Capitalization of FILE Records in the Audit Log

Valid on Windows

Audit records for FILE class records appear differently in the audit log in different releases of CA Access Control.

- In all r5 and r8 releases, the file path appears in lowercase.
- In r12.0 and r12.0 SP1, the file path is capitalized in the same way as the operating system represents the path on the computer.
- In r12.5 and later, the file path is capitalized in the same way as it appears in the CA Access Control FILE rule.

Example: Capitalization of FILE Records in the Audit Log

The following table shows how the audit records appear in the audit log for each CA Access Control release for the file named C:\tmp\TeSt.txt, for which you create a FILE record named C:\TMP\TEST.txt:

Release	Appearance in Audit File
r5 and r8	C:\tmp\test.txt
r12.0 and r12.0 SP1	C:\tmp\TeSt.txt
r12.5 and later	C:\TMP\TEST.txt

Appendix B: Trace Messages

This section contains the following topics:

Conventions

All messages begin with a date and time prefix, followed by an event-type word in uppercase and a symbol such as *:*, *!*, or *>*. The following table explains the meaning of the symbols.

:

CA Access Control was signaled for an event or performed an action.

>

CA Access Control made an authorization decision resulting in *D* (Deny), *P*, (Permit), or *BYPASS* (The event did not require the interpretation of an access rule-for example, a setuid request to the same UID as the current UID.)

!

CA Access Control detected an error-for example, a request from an unknown process.

Messages

The symbols described in the previous section precede the event arguments, described in this section.

ACTION : CA Access Control killed P=ppp

CA Access Control denied a setuid or login request and killed the requesting process (ppp) as a precautionary measure.

ALARM ! Uid uuu breached the system!!!

An unknown process made a request such as fork, exec, or setuid. The process is unknown to CA Access Control and, in addition, the UID assigned to the process is not assigned to any other process in the system. This implies that the user logged in without CA Access Control being notified. This situation can occur as a result of a software bug or if the user logged in immediately after CA Access Control scanned the current process status but before completing initialization.

APIAUTH ! P=ppp U=uuu ChangePasswd(user) Error 0xerr

Process *ppp*, associated with user *uuu*, wants to change the password of *user*. The result of this request was an error with its code specified in hex. Use the semsgtool utility to determine the nature of the error.

APIAUTH ! P=ppp U=uuu CheckPasswd(user) Error 0xerr

Process *ppp*, associated with user *uuu*, wants to check the validity of a new password for *user*. The result of this request was an error with its code specified in hex. Use the semsgtool utility to determine the nature of the error.

APIAUTH ! P=ppp U=uuu Error, Unknown API Service nnn

Process *ppp* used the Application Interface and passed a service code that the CA Access Control Programming Interface does not support, probably because of user error. Check the cause of the error, correct the source, and recompile it.

APIAUTH ! P=ppp U=uuu GeneralResourceProc Error nnn >description

Process *ppp*, working under UID *uuu*, issued a request to access a general resource; however, the specified resource cannot be resolved. Either the specified class is not defined or the specified access is not known, probably because of user error. Check your code, correct it, and recompile.

APIAUTH ! P=ppp U=uuu in VerifyCreate only for ROOT

Process *ppp*, working under UID *uuu*, issued a VerifyCreate request to build an ACEE. This operation is permitted only to multiuser processes that are associated with UID 0 (root).

If the specified process is to run as a multiuser process, rerun the process under root authorities. If not, determine why the process issued the request.

APIAUTH : P=ppp U=uuu in VerifyDelete only for ROOT

Process *ppp*, working under UID *uuu*, issued a VerifyDelete request to remove an ACEE. This operation is allowed only to multiuser processes that are associated with UID 0 (root).

If the specified process is supposed to run as a multiuser process, rerun it under root authorities. If not, determine why the request was issued.

APIAUTH ! P=ppp U=uuu LoginProc Error nnn >description

Process *ppp*, working under UID *uuu*, requested to verify a user's login. The CA Access Control login verification procedure failed. Contact your vendor's technical support.

APIAUTH ! P=ppp U=uuu NULL ACEE Error VerifyCreate (ACEEH=hhh)

A user process marked as “server” made a request to create an ACEE (probably as the server process was handling login for an accessor). The result is a NULL ACEE for one of the following reasons:

- The specified user is not defined in the CA Access Control database.
- The issuer of the VerifyCreate request did not provide all the information correctly.
- The specified user is not allowed to log in.

APIAUTH ! P=ppp U=uuu NULL ACEE Error VerifyDelete (ACEEH=hhh)

Process *ppp*, associated with user *uuu*, and which is probably marked as a 'server' process, has requested to delete the ACEE handle *hhh* (probably as part of handling the user's signoff). However, no ACEE is associated with this handle, so CA Access Control cannot delete it.

APIAUTH : P=ppp U=uuu Request with ACEEH=1 > New ACEEH=hhh

Process *ppp*, working under UID *uuu*, requested access to a general resource and supplied an ACEE handle of -1. CA Access Control used the ACEE handle associated with the requesting process. This message is typical of single user processes that request access to a resource. No action is required.

APIAUTH ! P=ppp U=uuu VerifyCreate(ACEEH=hhh) Error nnn

Process *ppp*, working under UID *uuu*, issued a request to VerifyCreate (to build an ACEE). The VerifyCreate procedure failed. Contact your vendor's technical support.

APIAUTH > P=ppp U=uuu VerifyCreate DENY (Result=[P/D/C]) string

The VerifyCreate request was denied for one of the following reasons:

- The specified user cannot login due to time or day rules
- The user cannot work from the specified terminal
- The specified password (if supplied) is incorrect
- One of the reasons described in the messages that follow.

APIAUTH > P=ppp U=uuu VerifyCreate OK (ACEEH=hhh)!

The VerifyCreate request was granted. An Accessor Environment Element (ACEE) was built in storage. CA Access Control returned an ACEE handle (ACEEH) to the calling program. If the specified user is not defined to CA Access Control, the function returned an ACEEH of -1.

APIAUTH ! P=ppp U=uuu VerifyDelete(ACEEH=hhh) [OK | Error 0xerr]

Process *ppp*, associated with user *uuu*, which is probably marked as a 'server' process, has requested the deletion of the ACEE handle *hhh* (probably as part of handling the user's signoff). The result of the VerifyDelete request is either OK or error; if the latter, the error code appears in hex as err. Use the utility semsgtool to determine the nature of the error.

APIAUTH > P=ppp U=uuu VerifyRequest(ACEEH=hhh, C=ccc, R=rrr, A=nnn) DENY (Result='D')Why ? detaileddenialreason

The request to access resource *rrr* of class *ccc* with access *xxx* was denied. If the ACEEH is -1, the denial was based on universal-access rules. If the ACEEH is not -1, the denial was based on the user associated with the specified handle. The second line provides a detailed reason for the denial.

APIAUTH > P=ppp U=uuu VerifyRequest(ACEEH=hhh, C=ccc R=rrr, A=xxx) PASS

The request to access a resource *rrr* of class *ccc* with access *xxx* was granted. If the ACEEH is -1 (the user is not defined to CA Access Control), the permission to access the resource was based on universal-access rules. If the ACEEH is not -1, the permission was based on access rules relating to the user associated with the specified handle.

CONNECT : P=ppp U=uuu ACEEH=hhh from ipip:port1 to socket 6000 host=iiii

A request to open a window on host *iiii* (X-Terminal or station) was made by process *ppp* associated with UID *uuu*.

Note: The port number is always 6000; all other TCP/IP connection requests are ignored by CA Access Control.

CONNECT > P=ppp U=uuu from ipip:port1 to socket 6000 host=iiii BYPASS

CA Access Control bypassed the CONNECT request without interpreting access rules, because the program executing in process *ppp* is the registered XDM program.

CONNECT > Result: [P/D/C] P=ppp ACEEH=hhh TERM=tttWhy ? detaileddecisiontext

The CONNECT result is *D* (Deny) or *P* (Permit). The second line provides a reason for the decision.

ERROR ! Cannot fork. Errno nnn.

During initialization, CA Access Control forks a few times to become a daemon. The fork request failed with the specified error number.

If you cannot determine the cause of the problem, contact your vendor's technical support.

ERROR ! Exec of CA Access Control agent failed ddd

The Engine cannot start up the Agent daemon. Check that the seagent executable is located in the right place, usually *ACInstallDir*/bin/seagent. If this file exists in the correct location, report the problem to your vendor's technical staff. In the message text, *ddd* is the error number that CA Access Control received from the operating system when trying to execute seagent.

ERROR ! Failed to get memory for LOGIN programsERROR ! Failed to get memory for NFS devicesERROR ! Failed to get memory for PRIV programsERROR ! Failed to get memory for XDM programs

These messages imply a severe shortage of memory. Either your computer does not meet the minimum memory requirements to run CA Access Control, or a software bug was found. Contact your vendor's technical support.

ERROR ! Failed to get memory for PROC table

When seosd starts up, it must scan all the running processes to resolve all required information on each running process. seosd failed to allocate memory for this purpose; therefore, it terminates execution. This is caused by a severe memory shortage.

ERROR ! Failed to register login pgm: programname

During startup, CA Access Control registers all executable files that are to be treated as login programs. The list of login programs is defined in the CA Access Control code for each operating system environment.

The specified *programname* cannot be located on the file system during startup. CA Access Control ignores the program and startup continues.

ERROR ! Failed to register privileged pgm: programname

During startup, CA Access Control registers all executable files that are to be treated as privileged programs. The specified *programname* cannot be located on the file system during startup. CA Access Control ignores the program and startup continues.

The list of privileged programs is defined in the CA Access Control code for each operating system environment.

ERROR ! Failed to register XDM pgm: programname

During startup, CA Access Control registers all executable files that are to be treated as XDM programs. The list of XDM programs is defined in the CA Access Control code for each operating system environment.

The specified *programname* cannot be located on the file system during startup. CA Access Control ignores the program and startup continues.

ERROR : No Memory for FileDb List

During startup, seosd cannot allocate memory to hold the list of protected files. This is probably due to a severe shortage of memory. The seosd daemon is terminated.

ERROR ! No Memory for GroupDb ListERROR ! No Memory for HostDb ListERROR ! No Memory for ServDb ListERROR ! No Memory for UserDb List

These messages imply a severe shortage of memory. Either your computer does not have the minimum memory required to run CA Access Control, or a software bug was found. Contact your vendor's technical support.

ERROR ! PreMatureExec Assuming FORK Child=ppp Parent=PPP

This message indicates that process ID (*ppp*) issued an EXEC system call, which is not known to seosd. Normally, such messages indicate that seosd was not yet informed of the FORK system call that preceded the EXEC request. It may indicate a problem in the serialization locks that the CA Access Control extension to the UNIX kernel, SEOS_syscall, must maintain.

If the *ppp* in the message text is the pid of seagent, you can ignore the message. If you get the message more than once, report the problem to your vendor's technical support.

ERROR ! P=ppp Exec Failed

CA Access Control received an EXEC event, but the inode number of the executable was zero. This message occurs when invoking a script file that does not contain the #! shell-program declaration line at the beginning. No action is necessary.

ERROR ! CA Access Control file table set failed

seosd attempted to set the file table (a table of all CA Access Control protected files); however, SEOS_syscall refused this request. The most likely causes are insufficient memory in the kernel, or different versions of seosd and SEOS_syscall. CA Access Control file protection cannot continue to function properly.

If you can, resolve the version mismatch. If everything looks fine, report the problem to your vendor's technical support.

ERROR ! seosini_ShutDown rv=errorno

CA Access Control encountered an error during shutdown. Report the error to your vendor's technical support.

ERROR ! String too general 'path'

An attempt was made to define a generic rule for file protection, probably through a newfile or newres FILE command. However, the specified path cannot be a generic file access rule. The file rule is not defined.

ERROR ! Unknown request: Type:ttt Pid=ppp, Buff=bbb

CA Access Control received a request from its system call, but the request type *ttt* is not recognizable. This can be due to a software version mismatch between the CA Access Control system call and seosd, or because of a software error. The request came from process *ppp*, and *bbb* is a printout of the request buffer. Report the problem to your vendor's technical support.

EXEC : P=ppp U=uuu G=ggg (D=ddd I=iii) Pgm:ProgramName [Attached to: ipaddress]

CA Access Control received a program execution event from process *ppp* associated with UID *uuu* and GID *ggg*. (A *ggg* value of -1 indicates that CA Access Control has not yet registered the GID of that process). In the message text, *ddd* and *iii* are the file's device number and inode, respectively. *Program-Name* is the zero argument used when invoking the program. The specified program is a regular program (that is, not setuid or setgid); therefore, CA Access Control grants its execution without invoking the database access rule decision mechanism. If the *ip-address* to which the process is attached is extractable, CA Access Control reports this in the message text.

EXEC sg: P=ppp U=uuu G=ggg (D=ddd I=iii) Pgm:ProgramName[Attached to: ipaddress]

CA Access Control received a program execution event from process *ppp* associated with UID *uuu* and GID *ggg*. (A *ggg* value of -1 means CA Access Control has not yet registered the GID of that process). In the message text, *ddd* and *iii* are the file's device number and inode, respectively. *Program-Name* is the zero argument used when invoking the program. The specified program is a setgid program; CA Access Control determines whether to grant its execution by invoking the database access rule decision mechanism. If the *ip-address* to which the process is attached is extractable, CA Access Control reports this in the message text.

EXECsu : P=ppp U=uuu G=ggg (D=ddd I=iii) Pgm:ProgramName[Attached to: ipaddress]

CA Access Control received a program execution event from process *ppp* associated with UID *uuu* and GID *ggg*. (A *ggg* value of -1 means CA Access Control has not yet registered the GID of that process). In the message text, *ddd* and *iii* are the file's device number and inode, respectively. *Program-Name* is the zero argument used when invoking the program. The specified program is a setuid program; CA Access Control determines whether to grant its execution by invoking the database access rule decision mechanism. If the *ip-address* to which the process is attached is extractable, CA Access Control reports this in the message text.

EXECsusg: P=ppp U=uuu G=ggg (D=ddd I=iii) Pgm:ProgramName[Attached to: ipaddress]

CA Access Control received a program execution event from process *ppp* associated with UID *uuu* and GID *ggg*. (A *ggg* value of -1 means CA Access Control has not yet registered the GID of that process). In the message text, *ddd* and *iii* are the file's device number and inode, respectively. *Program-Name* is the zero argument used when invoking the program. The specified program is a setuid and setgid program; CA Access Control determines whether to grant its execution by invoking the database access rule decision mechanism. If the *ip-address* to which the process is attached is extractable, CA Access Control reports this in the message text.

EXEC > P=ppp U=uuu (R=rrr E=eee S=sss) to (E=EEE) BYPASS

Although the program is setuid, setgid, or both, and its execution should have invoked the access rule decision mechanism, CA Access Control bypassed this check because the owner of the file *EEE* is the same as the current effective UID (*eee*). The program execution cannot change the scope of the privileges of the process. If the program is defined in the database as a trusted program and was modified or otherwise tampered with, program execution is not granted.

EXEC > Result: 'R' [stage=sss gstag=ggg ACEEH=hhh rv=rc]Why? DetailedDecisiontext

CA Access Control checked the authority of the user to execute the program and the result *R*, where *R* is either D (deny) or P (permit). The stage *sss* and the granting-stage *ggg* indicate which phase of the decision flow determined the result. The ACEE handle *hhh* was used as the accessor to the program. If the result is 'C' (check) it means CA Access Control did not make a decision, probably because of a software error-contact your vendor's technical support and provide them with the return value *rc*. *Detailed-Decision-text* is a textual description of the stage and granting-stage. If the result was *P*, the program is executed successfully. If the result is *D*, the program will not be executed and the user receives a permission denied message.

EXECARGS: 'execution arguments'

Because of an EXEC syscall, CA Access Control displays the executed command line with all the arguments passed to it.

EXIT : Going down...

CA Access Control started the shutdown process and disabled the interception of system calls.

FATAL ! in seosrt_InitDatabase (nnn) Layer = nnn Stage = nnn Return Code = 0xnnn

CA Access Control cannot initialize the database I/O routines. The possible reasons are:

- No CA Access Control database in the directory is identified by the dbdir token in the seos.ini file.
- The user invoking CA Access Control is not root.
- The database is corrupt.

If you cannot correct the problem, contact your vendor's technical support.

FILE : P=ppp U=uuu (D=dev I=inode) acc : pathname

Process *ppp* associated with userid *uuu* attempted to access a CA Access Control protected file. In the message text, *dev* and *inode* are the device and inode of the file being accessed, respectively; *acc* is the access mode (that is, READ, WRITE, and so on); and *pathname* is the real path name of the file being accessed.

FILE > Result 'D' CA Access Control File Only 'filename'

The result of the file access request is D (denial) because only CA Access Control can access this file. Even if the access rules permit access, CA Access Control is hard-coded to deny access to this file.

FILE > Result: 'R' [stage=sss gstag=gs ACEEH=hhh rv=rv (recordname)Why? detailedreasontext

The result R of the file access request is either D (deny) or P (permit). The stage *sss* and granting stage *gs* are mapped to a text-string reason, on the second line (following “Why?”). In the message text *hhh* is the accessor handle associated with the request's accessor and *record-name* is the name of the access rule record that triggered the decision to deny or permit access.

FORK : P=ppp U=uuu G=ggg Child=cppp Pgm:ProgramName

CA Access Control intercepted a fork request made by process *ppp* associated with UID *uuu* and GID *ggg.* The child process id is *cppp*. *Program-Name* is the program running in the parent process (and, initially, also in the child process). CA Access Control never denies a fork request; it is always granted. Variations of the fork system call, such as vfork and kfork, are also reported as fork requests.

GETCRED : P=ppp, Get Credentials by Ticket

This is an information-only message, which indicates that *ppp* (usually the process ID of the Policy Model daemon, sepmdd) requested the credentials of a specific ticket holder (a client process that requests the services of sepmdd). For more information, see the description of GTICKET in this appendix, and the description of sepmdd in the chapter "Utilities in Detail."

GPEERNAM: P=ppp, ADDR=addr, N=desc

CA Access Control intercepted the getpeername() system call to verify which IP address is associated with the current process. This system call is always granted. In the message text, *ppp* is the process id issuing the getpeername() call and *addr* is the IP address associated with the socket descriptor *desc.*

GTICKET: P=ppp, Get Authentication Ticket

This is an information-only message, which indicates that *ppp* requested seosd to issue an authentication ticket for it. Whenever the Policy Model client, sepmd, communicates with sepmdd, the server verifies the identity of the client through the passed ticket. The client sends the acquired ticket to the server using socket communication. The server then passes this ticket to seosd to get the credentials of the ticket holder with the GETCRED request. In this way, sepmdd ensures the identity of the client requesting its services.

INET : P=ppp, from ipaddress:localport to port portnumber

CA Access Control intercepted an incoming Internet accept request that was issued by the remote *ip-address* requesting the TCP/IP service *port-number*.

INET > Result: 'R' ipaddr>locport, stg=stage gtsg=gstageWHY ? DetailedReasonText

The result *R* of the Internet request is P (permit) or D (deny). In the message text, *ip_addr* is the IP address of the request. *Detailed-Reason-Text* is the textual description that indicates which stage and granting stage phase of the decision flow made the final decision to deny or allow the TCP/IP service for the requesting host.

INFO : AutoDisabling Tracedue to tight fsspace (space)

The trace facility automatically disables itself when the amount of free space left in the file system where the trace file resides, goes below a threshold specified by the trace_space_saver token in the seos.ini file. In the message text, *space* is the amount of free space left on the file system.

INFO : Can't fetch fs freespace (errno=err)

The Auto Disable feature of the trace facility cannot determine the amount of free space in the file system. In the message text, *err* is the error integer received from the UNIX statfs() call. Report the problem to your vendor's technical support.

INFO : DB Query

The seosd daemon received a request to extract information from the CA Access Control database.

INFO : DB Request

The seosd daemon received a request to modify or query data in the CA Access Control database.

INFO : Filter Mask: 'mask' is registered

The seosd daemon registers each filter mask that is read from the trcfilter.init file, so that messages matching the mask are not sent to the trace file.

INFO : GroupList Registered with nnn entries

When seosd runs under the NIS server, it caches all group entries (from /etc/group and NIS maps) at startup, so that seosd can solve GID to group name translations without invoking ypserv processes and TCP/IP requests. This message also indicates that the under_NIS_server token in seos.ini is set to YES. If the station where CA Access Control is running is not the NIS server, set the under_NIS_server token to NO. In the message text, *nnn* is the number of group entries that were cached.

INFO : HostList Registered with nnn entries

The seosd daemon caches all entries from /etc/hosts at startup. In the message text, *nnn* is the number of host entries cached.

INFO : Login program: programname is registered

The seosd daemon must recognize all the programs through which users log in to the system. CA Access Control treats a setuid system call invoked by a login program as a login request, and not as a setuid request. In the message text, *programname* is the full path of the login program that was registered. The seosd daemon takes the names of the login programs internally, from the CA Access Control startup code.

INFO : NFS Device Majors Registered, nnn entries

The checks that the Watchdog performs for trusted programs include checking the device number on which the file resides. This check can lead to errors if the file resides on an NFS mounted file system-especially an auto-mounted file system-for which device numbers can have a different value after boot. For this reason, CA Access Control registers the major device numbers of NFS file systems so that they can ignore the non-stable minor device number. CA Access Control has a list of major device numbers for NFS mounted file systems in each environment. If your installation uses a network mounted file system that CA Access Control does not recognize, contact your vendor's technical support for instructions about adding major device numbers to the list. In the message text, *nnn* is the number of major device numbers registered as NFS mounted file systems.

INFO : P=ppp ended

Process *ppp* ended. seosd disassociates this process number from its ACEE (accessor environment element). If process *ppp* was the last process associated with its ACEE, (that is, no other parent processes or subprocesses use the same environment), then the ACEE is removed from storage. This message is not issued immediately after the process has terminated; it is issued only when CA Access Control performs some “garbage collection” to reuse process entries in its internal tables.

INFO : P=ppp Exec Failed

This message indicates that process *ppp* failed to execute the last EXEC syscall, because UNIX refused this request (after CA Access Control granted the execution). Therefore, CA Access Control restores the value of the former executable that was associated with this process, as the program running under this process ID. In most cases, the process terminates. This is not necessarily an error, and you need not take any special action. However, you should use UNIX tools to isolate the reason that execution failed. In most cases, the reason is that a shell script does not have the “#!/bin/sh” header on the first line.

INFO : P=ppp Unknown TTY type typename

The seosd daemon cannot determine if the process *ppp* is using a real TTY or a pseudo TTY. Contact your vendor's technical support.

INFO : Privileged program: programname is registered

The seosd daemon registers a few privileged programs. Such programs are allowed to setuid to any user without checking the SURROGATE class. Currently, you can only make /bin/sendmail a privileged program, due to its flow requirements. You must keep this list as small as possible; we recommended that seoswd monitor all privileged programs to make sure they remain trusted. In the message text, *programname* is the full path of the registered program.

INFO : Restricted File Table set with nnn entries

During startup, seosd found *nnn* entries for CA Access Control protected files, and successfully passed this list to the CA Access Control extension of the UNIX kernel. This is an information-only message.

INFO : SEOS_syscall UnRegister rc=nnn

During shutdown, seosd unregisters itself to the kernel so that it can start up again. In the message text, *nnn* is the return code, which should be zero. If the return code is not zero, report the problem to your vendor's technical support.

INFO : ServList Registered with nnn entries

The seosd daemon caches all entries from /etc/services at startup. In the message text, *nnn* is the number of host entries that were cached.

INFO : ServList registered with nnn portmapper entries

While starting up, seosd registered nnn TCP/IP services that are resolved by the portmapper. This is an information-only message.

INFO : Set site

The seagent daemon, the CA Access Control daemon responsible for communication with other CA Access Control stations, sent seosd a connection request from a remote station.

INFO : Setting PV C=ccc O=ooo P=ppp

The seoswd daemon set the value of property *ppp* in object *ooo* of class *ccc*.

INFO : UserList Registered with nnn entries

When seosd runs under the NIS server, it caches all user entries (from /etc/passwd and NIS maps) at startup, so that seosd can solve UID to user name translations without invoking ypserv processes and TCP/IP requests. This message also indicates the under_NIS_server token in seos.ini is set to YES. If the computer where CA Access Control is running is not an NIS server, set under_NIS_server token to NO in seos.ini. In the message text, *nnn* is the number of user entries that were cached.

INFO : XDM program: programname is registered

XDM programs are those programs that display the userid and password box on X-terminals. XDM programs run under *superuser*, who usually cannot open windows on X-terminals. However, the XDM program must open a window on an X-terminal to present a box with the userid and password for the user to specify. seosd therefore bypasses terminal checking if the program issuing the CONNECT request is a registered XDM program.

KILL : P=ppp U=uuu kill [Process | All Except] (nn): (proclist)

Process *ppp* associated with user *uuu* attempted to kill all the processes listed in *proclist* (or all the processes except the processes in the list). In the message text, *nn* is the number of target processes.

KILL > Result 'R' [stage=sss gstag=gs rv=rr] ACEEH=hhhWhy? detailedreasontext

The result R of the kill event is either D (deny) or P (permit). In the message text, *sss*, *gs*, and *rr* are the stage, granting stage, and return value of the CA Access Control decision routines, and *hhh* is the accessor handle associated with the kill event. The *detailed-reason-text* appears in the second line and is a derivation of the stage and granting stage codes.

LOGIN : P=ppp User=uuu Terminal=ttt

The seosd daemon intercepted a login request from user *uuu* working on terminal *ttt* under process number *ppp*. A Login Result message should follow this message.

LOGIN > Result: 'R' [stage=stage gstag=gstage rv=nnn] ACEEH=hhh[Why ?detaileddenialreason]

The result of the login request R is either D (deny) or P (permit). In the message text, *stage* and *gstage* are numbers indicating the phase in the CA Access Control flow that made the decision to grant or deny the login request. If the login was permitted, *hhh* is the ACEE handle that is now associated with the issuing process. If the login was denied, *hhh* is set to -1 and a *detailed-denial-reason* appears in the second line. If the *detailed-denial-reason* relates to resource access (such as “no rule granting access to resource”), the resource in question is the terminal from which the user issued the login request.

LOGIN > Result: 'D' Login Disabled for ALL

The login request was denied because login is currently disabled for all users.

LOGIN > Result: 'D' Login Disabled for U=uuu

The login request was denied because login is currently disabled for the specific user. The reason can possibly be that this user is already logged in.

MESSAGE : string

A marker message is placed in the trace file by console request.

NEWPASS : Set new password

The sepass utility requested to set a new password for a userid.

PW_ATTCK: P=ppp make nnn attempts in sss seconds from terminal

The seosd daemon detected that process *ppp*, which is running one of the registered login programs, made *nnn* attempts to specify a user/password combination with no success. CA Access Control concluded that a password guess attack originated at the terminal specified in the message text, and wrote an audit record to the CA Access Control audit file. PWATTACK audit records can trigger actions by the log routing daemons (selogrcd and selogrd).

RESTART : DBSERV restarted by Watchdog (P=ppp)

The seoswd daemon has restarted seosd. In the message text, *ppp* is the process ID of seosd.

SCONSOLE: Login Disabled For UID: uuu

The CA Access Control console utility, secons, issued a request to disable a login request for the userid *uuu*. From this point, login requests for the specified userid are denied.

SCONSOLE: Login is already Disabled for U=uuu

The secons utility issued a request to disable login request for the userid *uuu*. However, login is already disabled for this userid.

SCONSOLE: Login is not Disabled for U=uuu

The secons utility issued a request to re-enable login for the userid *uuu*. However, login is already enabled for this userid.

SCONSOLE: Login Is Now Disabled

The secons utility issued a request to disable login for all users. From this point on, login requests by any user are denied.

SCONSOLE: Login Is Now Enabled

The secons utility issued a request to re-enable login for all users. From this point on, login requests are allowed.

SCONSOLE: Login ReEnabled for U=uuu

The secons utility issued a request to re-enable login for a specified user. From this point on, login requests for this specific user are allowed.

SCONSOLE: No more space in Disabled Logins Table

The secons utility issued a request to disable login for a particular user. However, the login disable table is full. Contact your vendor's technical support.

SCONSOLE: U=uuu is not allowed for operation

A user without the OPERATIONS attribute tried to use one of the secons switches that are not allowed for non-OPERATIONS users.

SCONSOLE: U=uuu is not allowed to disable login for U=uuu2

The user *uuu* tried to disable login for user *uuu2* through secons. However, only root and user *uuu2* are allowed to disable login for *uuu2*.

SCONSOLE: U=uuu is not allowed to Reenable login for U=uuu2

The user *uuu* tried to re-enable login for user *uuu2* through secons. However, only root and *uuu2* are allowed to re-enable login for *uuu2*.

SETGRPS : P=ppp to grouplist

The process *ppp* issued the setgroups system call for the groups specified in *grouplist*.

SGID : P=ppp U=uuu G=ggg to GGG (GROUP.groupname) ACEEH=hhh D=devnum I=inode

Process *ppp*, running with the authorities of UID *uuu* and GID *ggg*, issued a setgid system call for the GID *GGG*. CA Access Control checks the authority of that process using the SURROGATE class and object GROUP.*groupname*, and uses *hhh* as the accessor handle for the request. In the message text, *devnum* and *inode* are the device and inode of the issuing program, respectively. A “SGID Result” message should follow this one.

SGID > P=ppp U=uuu (RG=rg EG=eg SG=sg) to (RG=trg EG=teg SG=tsg) () BYPASS

CA Access Control granted the setgid request without checking any SURROGATE access rules. In the message text, *ppp* is the issuing process id; *uuu* is the userid associated with this process; *rg*, *eg*, and *sg* are the real, effective, and saved GID of that process; and *trg*, *teg*, and *tsg* are the target effective, real, and saved GID with which the setgid request was issued. The reason for the bypass is usually because the current real or saved GID is the same as the target GID, and therefore the setgid request does not change the security scope of the user.

SGID > Result: 'R' [stage=stage gstag=gstage ACEEH=hhh]Why? detailedreasontext

CA Access Control checked the setgid request against a SURROGATE access rule and the result R is P (permit) or D (deny). The decision was made on behalf of the accessor handle *hhh*. In the message text, *detailed-reason-text* is the reason for the denial or grant.

SHUTDOWN! Request Denied. U=uuu not allowed to SHUTDOWN the Server

The userid *uuu* tried to shut down seosd using secons; however, this user's profile does not have the OPERATIONS attribute. The request was therefore denied.

SHUTDOWN: Server going down upon operator's request

The seosd daemon started shutting down following a request from an authorized operator.

SHUTDOWN: Terminating CA Access Control daemon daemonname P=ppp RV=nnn

CA Access Control terminated its daemon *ppp* as part of its shutdown process; CA Access Control also shuts down seoswd and seagent.

STARTUP: CA Access Control daemon PID=ppp

The seosd daemon was started; its process ID is *ppp*.

STREAM c: P=ppp Closes Stream Id=iii

Process *ppp* closed a stream with stream ID *iii*. CA Access Control keeps track of all stream-open and stream-close operations to determine later-when a TCP/IP request is processed on behalf of a specific stream-id-which process ID owns the stream.

STREAM o: P=ppp Opens Stream Id=iii

Process *ppp* opened a stream with stream ID *iii*. CA Access Control keeps track of all stream-open and stream-close operations to determine later-when a TCP/IP request is processed on behalf of a specific stream-id-which process ID owns the stream.

SUID > P=ppp U=uuu (R=r E=e S=s) to (R=tr E=te S=ts) (reason) BYPASS

CA Access Control granted the setuid request without checking any SURROGATE access rules. In the message text, *ppp* is the issuing process id; *uuu* is the userid associated with this process; *r*, *e*, and *s* are the real, effective and saved UIDs of process *ppp*; and *tr*, *te*, and *ts* are the target effective, real, and saved UIDs with which the setuid request was issued. The reason for the bypass is usually because the current real or saved UID is the same as the target UID, and therefore the setuid request does not change the security scope of the user. Other possible reasons are that the program issuing the setuid system call is a privileged program (in which case *reason* is For Priv), or that the issuing program is a login program that switches UIDs several times before and after the actual login (in which case *reason* is specified as For Login).

SUID : P=ppp U=uuu (R=r E=e S=s) to USER.username (R=tr E=te S=ts)D=devnum I=inode

Process *ppp*, running with the authority of userid *uuu*, issued a setuid system call to change the current real, effective, or saved UID to UID *uuu*. CA Access Control checks the authority of that process using the SURROGATE class and object USER.*usernam* for that request. In the message text, *devnum* and *inode* are the device and inode of the issuing program, respectively. A “SUID Result” message should follow this one.

SUID > Result: 'R' [stage=stage gstag=gstage ACEEH=hhh rv=rv]Why? detailedreasontext

CA Access Control checked the setuid request against a SURROGATE access rule and the result R is P (permit) or D (deny). The decision was made on behalf of the accessor handle *hhh*. In the message text, *detailed-reason-text* is the reason for the denial or grant.

VERPASS : Verify password

CA Access Control received a request to verify password validity for a user.

WAKE_UP : Server going up

The seosd daemon started to initialize.

WARNING : Associate P=ppp ACEEH=hhh

CA Access Control performs an association between a process and an accessor handle (ACEEH) for any fork request. This message indicates that the association cannot be performed, either because the handle *hhh* is -1 or because *hhh* is not a valid accessor handle. In the latter case, contact your vendor's technical support.

WARNING : Can't verify P=ppp

This message follows an Unknown P= message that indicates a fork request from an unknown process. CA Access Control tries to determine who the user is that UNIX associates with that user. This verification task cannot be completed. A possible reason is that the process has already terminated. If this is not the case, contact your vendor's technical support.

WARNING : DeAssociate P=ppp ACEEH=hhh

CA Access Control performs a dissociation between a process and an accessor handle (ACEEH) for any process that is terminated. This message indicates that the dissociation cannot be performed, either because the handle *hhh* is -1 or because *hhh* does not exist as a valid accessor handle. In the latter case, report the problem to your vendor's technical support.

WARNING : ExecArg for entry with P=ppp not NULL

This warning appears when CA Access Control finds a new process that was not known to the system, and for which the executing program is not known. In most cases, you can ignore the message. If the system does not produce the expected results, contact your vendor's technical support.

WARNING : Failed to get ACEEH of P=ppp

CA Access Control was requested to check the authority of process *ppp* but there was no valid accessor handle for that process. In most cases, the reason is that the user associated with the process is not a CA Access Control defined user, or that the process is unknown to the CA Access Control system. In both cases, CA Access Control gives this process only universal access rights. If the system does not produce the expected results, contact your vendor's technical support.

WARNING : Login for P=0 ???

When this message appears during startup in systems other than AIX, you can ignore it. If it appears during normal work (after seosd is started and functions), or during startup under AIX, then it identifies a software error, in which case you should contact your vendor's technical support.

WARNING : CA Access Control failed to kill P=ppp reason=nnn

As a measure of caution, CA Access Control kills processes trying to get sensitive privileges that may create loopholes. Such events can be attempts to surrogate the UID (setuid system-call) with no permission. CA Access Control attempted to kill the violating process, but failed to do so. The reason for the failure is detailed in the reason code returned by the kill system call.

WARNING : Terminal for entry with P=ppp not NULL

This warning appears when CA Access Control finds a new process that was not known to the system and for which the executing program is not known. In most cases, you can ignore the message. If the system does not produce the expected results, contact your vendor's technical support.

WARNING : Unknown P=ppp

This message indicates a fork request that was issued by a process not known to CA Access Control. If this message appears for seoswd or seagent during startup, you can ignore it. At other times, it can imply a software error because CA Access Control cannot verify the actual authority of that process. For the latter case, contact your vendor's technical support.

WATCHDOG: Ask if I'm Here (AYT)

The seoswd daemon tried to verify whether seosd is alive and give the expected response. In the message text, AYT is the seoswd “are you there” challenge. You can and should ignore this message; filter it out using the trcfilter.init file. The message implies normal behavior of seoswd.

WATCHDOG: Init initializationtext

The seoswd initialization message, which you can ignore.

WATCHDOG: Log logtext

The seoswd daemon issued a log request. The log request is detailed in *log-text.*

WATCHDOG: SecFile operation result

The seoswd daemon requested the daemon to extract information regarding secured files. In the message text, *operation* can be GETFIRST or GETNEXT; the result can be OK if such information was extracted, or NOFOUND if there are no more secured files in the CA Access Control database. This message signifies normal behavior of seoswd to scan secured files.

WATCHDOG: Timer

The seoswd daemon issues a timer request every few seconds (as set by the seos.ini file). You can and should filter out this message using the trcfilter.init file.

WATCHDOG: Trust Pgm: programname [OK | NOTOK]

The seoswd daemon marked the specified program as a trusted program. This implies that the specified program passed the digital signature tests. In the message text, OK means the trust operation completed successfully, and NOTOK means that seoswd failed to mark the program as trusted. The reason for NOTOK is probably a corrupted database, in which case you should contact your vendor's technical support.

WATCHDOG: Untrust Pgm: programname [OK | NOTOK]

The seoswd daemon marked the specified program as untrusted. This implies that the specified program did not pass the digital signature checks of seoswd. In the message text, OK means that the untrust operation has completed successfully, and NOTOK means that seoswd failed to mark the program as untrusted. A possible reason for NOTOK can be a corrupted database, in which case you should contact your vendor's technical support.

Appendix C: String Matching

This section contains the following topics:

Wildcard Expressions

This section describes the syntax that can be used to build wildcard expressions.

CA Access Control performs string matching (globbing) using the wildcard matching and character lists.

Wildcard Matching

CA Access Control supports the following wildcard characters:

Character	Match
* (asterisk)	Any sequence of zero or more characters.
? (question mark)	Any single character.

Character Lists

A character list enclosed by square brackets ([ ]) can contain one or more characters. CA Access Control uses these characters as positive or negative matching criteria.

A character list can be composed of one or more characters. For this type of list, CA Access Control matches any single character in the list. If the list within the brackets is preceded by a caret (^), CA Access Control matches any single character, which is *not* in the list.

A range is a type of character list that specifies a range of characters. CA Access Control matches all the characters in the list, inclusively. If a caret (^) precedes the list, CA Access Control excludes all the characters in the specified list. You can specify both ends of the range, or only its first or last character.

The following table describes the character lists that can be used. Remember, in this syntax, you include the square brackets. Each of the expressions *ch1*, *ch2*, and *chN*, stands for a single character.

List	Description
[*ch1ch2...chN***]**	CA Access Control matches any single character in the list enclosed by the square brackets.
[^*ch1ch2...chN***]**	CA Access Control matches any single character that is *not* in the list enclosed by the square brackets.
[*ch1-ch2***]**	CA Access Control matches any single character in the range, inclusive.
[^*ch1-ch2***]**	CA Access Control matches any single character that is *not* in the inclusive range.
[-*ch2***]**	CA Access Control matches any single character with an ASCII value lower than or equal to the specified character (*ch2*).
[^-*ch2***]**	CA Access Control matches any single character with an ASCII value equal to or higher than the specified character (*ch2*).
[*ch1-***]**	CA Access Control matches any single character with an ASCII value equal to or higher than the specified character (*ch1*).
[^*ch1-***]**	CA Access Control matches any single character with an ASCII value equal to or lower than the specified character (*ch1*).

Examples: Wildcard Matching

To make a single character a “don't care” character that matches any other single character, use a question mark (?):

Specify	To match
mmc?	mmc3, mmcx, mmc5
mmc?.t	mmc1.t, mmc2.t
mmc04.?	mmc04.a, mmc04.1

To match any string of zero or more characters, use an asterisk (*):

Specify	To match
i.c	main.c, list.c, and so on

Specify	To match
st*.h	stdio.h, stdlib.h, string.h, and so on
*	All records of the specified class

To match any character in a list, follow one of these examples:

Specify	To match
[abcgk]	a, b, c, g, or k
[^abcgk]	Any character other than a, b, c, g, or k, such as A, B, d, e, f, or @.
[a-z]	Any character between a and z, inclusive.
[^a-z]	Any character with an ASCII value less than “a” or greater than “z.”
[Z-]	Any character with an ASCII value greater than Z's, such as a, b, \, or ~.
[^-A]	Any character with an ASCII value *not* lower than A's, such as B, a, c, or ~.

Appendix D: Used Ports

This section contains the following topics:

CA Access Control UNIX Endpoint Used Ports

CA Access Control uses the following TCP ports on UNIX by default:

Port Number	Description	Direction	Source	Target	Comments
8891	CA Access Control Client Applications	Incoming	Remote CA Access Control Utilities	CA Access Control Agent	You can change the default port number by modifying the /etc/services file settings. To modify the default port number, add the following line, then restart CA Access Control daemons: `seoslang2 port-number/ tcp`
5249	SSL Communications	Incoming	Remote CA Access Control Utilities	CA Access Control Agent	FIPS 140-2 compliant. For more information about SSL communication, see the SSL, Authentication, and Certificates section in the *Implementation Guide*.

Port Number	Description	Direction	Source	Target	Comments
8892	Starting seosd from a remote computer	Incoming		seosload	When seload loads daemons on a remote computer, inetd (internet services daemon) on the remote computer executes the rseloadd program. This program executes seload locally and exits; it receives the parameters on this port. You can change the default port number by modifying the /etc/services file settings. To modify the default port number, add the following line, then restart CA Access Control daemons: `seosload` *`port-number`*`/ tcp` **Note:** The communication on this port is not encrypted since it does not send any sensitive information.
7443	Reports and Audit Events	Outgoing	ReportAgent	Distribution Server	
8891	CA Access Control Client Applications	Outgoing	Policyfetcher	Distribution Server	Distributing AC policies to endpoints through Advanced Policy Management.
5249	SSL Communications	Outgoing	Policyfetcher	Distribution Server	Distributing AC policies to endpoints through Advanced Policy Management when SSL is enabled.

CA Access Control Windows Endpoint Used Ports

CA Access Control uses the following TCP ports on Windows by default:

Port Number	Description	Direction	Source	Target	Comments
8891	CA Access Control client applications	Incoming	selang.exe, sepmdd.exe (PMD), eACSigUpdate.exe, SegraceW.exe (grace login and password settings), secons.exe (remote shutdown and IP address refresh), policydeploy.exe, devcalc.exe, policyfetcher.exe	CA Access Control Agent	You can change the default port number by modifying the %SystemRoot%\drivers\etc\services file settings. To change the default port number, add the following line, then restart CA Access Control services: `seoslang2 port-number/tcp`

Port Number	Description	Direction	Source	Target	Comments
5249	SSL Communications	Incoming		CA Access Control Agent	FIPS 140-2 compliant

Port Number	Description	Direction	Source	Target	Comments
7443	Reports and audit events	Outgoing	ReportAgent	Distribution Server	
8891	CA Access Control Client Applications	Outgoing	Policyfetcher	Distribution Server	Distributing AC policies to endpoints through Advanced Policy Management.
5249	SSL Communications	Outgoing	Policyfetcher	Distribution Server	Distributing AC policies to endpoints through Advanced Policy Management when SSL is enabled.

Server Components Used Ports

CA Access Control uses the following TCP ports for its server components by default:

Port Number	Description	Direction	Source	Target
7243	Report snapshots using SSL	Outgoing	CA Access Control Enterprise Management Server	Distribution Server
5248	Local web-based interface communications	Outgoing	CA Access Control Enterprise Management	CA Access Control Web Service

In addition to these ports, you open the following ports:

- On the central databases' computer for communication with the CA Access Control Enterprise Management, if these are on a separate computer.
- On the Report Portal (BusinessObjects) computer to access the InfoView application from remote computers (8080 by default).
- On the CA Access Control Endpoint Management and CA Access Control Enterprise Management computer to access the web-based interfaces from remote computers (18080 by default).
- On the computer where you install Oracle Database to access the web-based interface from remote computers (by default, 8080 or 7443 for SSL).

UNIX Authentication Broker Used Ports

UNIX Authentication Broker uses the following TCP ports on UNIX by default:

Number	Description	Source	Target
88	Kerberos traffic	UNIX Authentication Broker Agent	Active Directory
389	Kerberized LDAP	UNIX Authentication Broker Agent	Active Directory
445	Microsoft directory services	UNIX Authentication Broker Agent	Active Directory
464	Kerberos kpasswd	UNIX Authentication Broker Agent	Active Directory
3268	Global Catalog	UNIX Authentication Broker Agent	Active Directory
7243	Report snapshots using SSL	Report Agent	Distribution Server

UNIX Authentication Broker uses the following UDP on UNIX by default:

Number	Description	Source	Target
53	DNS	UNIX Authentication Broker Agent	Active Directory
88	Kerberos traffic	UNIX Authentication Broker Agent	Active Directory
123	NTP	UNIX Authentication Broker Agent	Active Directory
389	Kerberized LDAP	UNIX Authentication Broker Agent	Active Directory
464	Kerberos kpasswd	UNIX Authentication Broker Agent	Active Directory

Privileged User Password Management Used Ports

CA Access Control uses the following TCP ports to manage privileged user passwords by default:

Number	Description	Direction	Source	Target	Comments
135	Remote Procedure Call	Incoming	Distribution Server	Windows Endpoints	Remote Procedure Call (RPC) needed for WMI.
445	Remote registry access	Incoming	Distribution Server	Windows Endpoints	Remote registry access that is needed for WMI.
139	Optional Port	Incoming	Distribution Server	Windows Endpoints	This port is required when Windows endpoint uses the NETBIOS protocol. WMI can use NETBIOS over port 139 in case of failure to use port 445 over TCP. If you did not configure the endpoint to use NETBIOS, you do not need to open port 139.
<WMI fixed port>	WMI communications	Incoming	Distribution Server	Windows Endpoints	Configure the endpoint with the WMI fixed port when configuring Active Directory endpoint only.
389	ADSI Communication	Incoming	ENTM	Windows Endpoint	This port is required for managing Windows endpoint

Number	Description	Direction	Source	Target	Comments
<ADSI fixed port>	ADSI communications		ENTM	Windows Endpoints	Configure the endpoint with the ADSI port.
22	SSH Port	Incoming	ENTM	SSH Endpoint or Network Device	This port is required for managing SSH devices through the SSH protocol.
	Telnet Port	Incoming	ENTM	SSH Endpoint or Network Device	This port is required for managing SSH devices through the Telnet protocol.
1521	Oracle database port	Incoming	ENTM	Oracle Endpoint	This port is required for managing Oracle endpoints.
1433	Microsoft SQL Server database port	Incoming	ENTM	Microsoft SQL Server Endpoint	This port is required for managing Microsoft SQL Server endpoints.
18080,18433	Optional Port	Incoming	Browser	ENTM	Use this port when using the ENTM web UI from a machine which is behind a firewall.

ObserveIT Used Ports

ObserveIT uses the following TCP ports:

Number	Description	Listener	Sender	Comments
4884	ObserveIT Agent	ObserveIT Application Server	ObserveIT Agent	By default, the communication is not SSL-enabled. Use port 443 if you enable SSL communication.
4884	ObserveIT Web Console	ObserveIT Application Server	ObserveIT Agent	By default, the communication is not SSL-enabled. Use port 443 if you enable SSL communication.
1433		Database Server	ObserveIT Application Server and ObserveIT Web Console	

UARM Used Ports

UARM uses the following TCP ports:

Number	Description	Listener	Sender	Comments
6789	Agent command and control listening port	ELM Agent	Agent Manager	The communication is encrypted using AES256 key.
17001	Default Secure Agent to UARM communication	Log Depot	ELM Agent	The communication is encrypted using the AES256 key.
17000	Event log store listening port	Log Depot	ELM Agent	The communication is encrypted using the AES256 key.

Number	Description	Listener	Sender	Comments
80	ELM Subscription	Akamai Server	Subscription Service	This communication is not encrypted.
80	ELM Browser Interface			This TCP communication is encrypted and is automatically redirected to port 5250.
443	ELM Browser Interface	Tomcat	ELM Server	This TCP communication is encrypted and is automatically redirected to port 5250.