CA Access Control

selang Reference Guide

r12.5 SP1

This documentation and any related computer software help programs (hereinafter referred to as the "Documentation") are for your informational purposes only and are subject to change or withdrawal by CA at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and may not be used or disclosed by you except as may be permitted in a separate confidentiality agreement between you and CA.

Notwithstanding the foregoing, if you are a licensed user of the software product(s) addressed in the Documentation, you may print a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO THE END USER OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2009 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

Third-Party Notices

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

(c) Copyright IBM Corporation 1999, 2002

All Rights Reserved

CA Product References

This document references the following CA products:

- CA Access Control Premium 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 Manager (formerly Unicenter Service Desk)
- 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.

Contact CA

Contact Technical Support

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

- 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

Provide Feedback

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

If you would like to provide feedback about CA product documentation, complete our short customer survey, which is also available on the CA Support website, found at http://ca.com/docs.

Documentation Changes

The following documentation updates have been made since the r12.5 release of this documentation:

- Access Authority by Class (see page 28)—Updated topic lists the valid access values for the WINSERVICE class.
- get devcalc Command—Retrieve Policy Deviation Data (see page 116)—Updated topic provides an additional example.
- start devcalc Command—Initiate Policy Deviation Calculation (see page 153)—Updated topic provides information about the -precise parameter.
- SUDO Class (see page 387)—Updated topic explains that you must set the interactive flag for a SUDO record for interactive Windows applications.

Contents

Chapter 1: Introduction 15

About this Guide 15
Who Should Use this Guide 15

Chapter 2: The selang Command Language 17

CA Access Control Command Line Interpreter 17
selang Utility—Run the CA Access Control Command Line 18
Features of the selang Command Shell 20
selang Command Syntax 25
selang Command Authorization 26
Access Control List Support 26
Access Authority by Class 28
Windows Access Authority by Class 31
selang Environments 33
selang Configuration on UNIX 34
Change the User File 35
Change the File for Updating Groups 35
Automatic Backup of the UNIX User and Group Files 35
Get selang Help 36

Chapter 3: selang Commands 37

selang Commands Reference 37
selang Commands in the AC Environment 42
alias Command—Define selang Aliases 42
authorize Command—Set Access Authorities on a Resource 44
authorize- Command—Remove Access Authorities from a Resource 49
check Command—Determine a User's Access Authority 52
checklogin Command—Determine Login Information 53
checkpwd Command—Check a Password for Compliance 54
chfile Command—Modify File Records 56
ch[x]grp Command—Change Group Properties 62
chres Command—Modify Resource Records 74
ch[x]usr Command—Change User Properties 90
deploy Command—Initiate Policy Deployment 109
deploy- Command—Initiate Policy Removal 110
editfile Command—Create and Modify File Records 111

edit[x]grp Command—Create and Modify Group Records....111
editres Command—Modify Resource Records....111
edit[x]usr Command—Modify User Records....111
end_transaction Command—Complete Recording Dual Control Transactions....112
environment Command—Set the Security Environment....112
find Command—List Database Records....113
get dbexport Command—Retrieve Exported Database Rules....114
get devcalc Command—Retrieve Policy Deviation Data....116
help Command—Get selang Help....118
history Command—Show Previously Issued Commands....119
hosts Command—Connect to a Remote CA Access Control Terminal....119
join[x] Command—Add Users to Internal Groups....122
join[x]- Command—Remove Users from Groups....125
list Command—List Database Records....126
newfile Command—Create File Records....126
new[x]grp Command—Create Group Records....126
newres Command—Create Resource Records....126
new[x]usr Command—Create User Records....127
rename Command—Rename a Database Record....127
rmfile Command—Delete File Records....128
rm[x]grp Command—Delete Group Records....129
rmres Command—Delete a Resource....130
rm[x]usr Command—Delete User Records....132
ruler Command—Select Properties to Display....133
setoptions Command—Set CA Access Control Options....135
search Command—List Database Records....143
showfile Command—Display File Properties....143
show[x]grp Command—Display Group Properties....145
showres Command—Display Resource Properties....147
show[x]usr Command—Display User Properties....149
source Command—Execute Commands from a File....151
start dbexport Command—Initiate Database Export....152
start devcalc Command—Initiate Policy Deviation Calculation....153
start_transaction Command—Start Recording Dual Control Transactions....155
unalias Command—Remove selang Aliases....157
undeploy Command—Initiate Policy Removal....158
selang Commands in the Remote Configuration Environment....158
editres config—Modify Configuration Settings....158
find config—List Configuration Resources....161
showres config—Display Configuration Information....162
selang Commands in the Native UNIX Environment....163
chfile Command—Modify UNIX File Settings....163
chgrp Command—Modify UNIX Groups....165

chusr Command—Modify UNIX Users 166
editfile Command—Modify UNIX File Settings 167
editgrp Command—Create and Modify UNIX Groups 167
editusr Command—Create and Modify UNIX Users 168
find file Command—List Native Files 168
join Command—Add Users to Native Groups 169
join- Command—Remove Users from Native Groups 170
newgrp Command—Create UNIX Groups 171
newusr Command—Create UNIX Users 171
rmgrp Command—Delete UNIX Groups 171
rmusr Command—Delete UNIX User 172
showfile Command—Display Native File Properties 173
showgrp Command—Display Native Group Properties 174
showusr Command—Display Native User Properties 175
selang Commands in the Native Windows Environment 176
authorize Command—Set Accessors' Authority to Access Windows Resources 176
authorize- Command—Remove Accessors' Authority to Access Windows Resources 178
chfile Command—Modify Windows File Settings 180
chgrp Command—Modify Windows Groups 181
chres Command—Modify Windows Resources 183
chusr Command—Modify Windows Users 186
editfile Command—Modify Windows File Settings 192
editgrp Command—Create and Modify Windows Groups 192
editusr Command—Create and Modify Windows Users 193
editres Command—Create and Modify Windows Resources 193
find file Command—List Native Files 193
find {xuser|xgroup} Command—List Enterprise Users or Groups 194
join Command—Add Users to Native Groups 195
join- Command—Remove Users from Native Groups 196
newgrp Command—Create Windows Groups 197
newres Command—Create Windows Resources 197
newusr Command—Create Windows Users 197
rmgrp Command—Delete Windows Groups 198
rmres Command—Delete a Windows Resource 198
rmusr Command—Delete a Windows User 199
setoptions Command—Set CA Access Control Windows Options 200
showfile Command—Display Native File Properties 202
showgrp Command—Display Native Group Properties 203
showres Command—Display Native Resource Properties 204
showusr Command—Display Native User Properties 205
xaudit Command—Modify System Access Control List 206
xaudit- Command—Remove System Access Control List 207
selang Commands in the Policy Model Environment 208

backuppmd Command—Back up a PMDB 209
createpmd Command—Create a PMDB on a Host 209
deletepmd Command—Remove a PMDB from a Host 211
findpmd Command—List PMDBs on the Host 212
listpmd Command—List Information about a PMDB 212
pmd Command—Control a PMDB 213
restorepmd Command—Restore a PMDB 215
subs Command—Add Subscribers or Subscribe Databases 216
subspmd Command—Change Parent PMDB 217
unsubs Command—Remove a Subscriber 217

Chapter 4: Classes and Properties 219

Class and Property Information 219
Classes in the AC Environment 220
ACVAR Class 220
ADMIN Class 222
AGENT Class 226
AGENT_TYPE Class 227
APPL Class 228
AUTHHOST Class 234
CALENDAR Class 238
CATEGORY Class 240
CONNECT Class 241
CONTAINER Class 245
DEPLOYMENT Class 250
DICTIONARY Class 255
DOMAIN Class 256
FILE Class 260
GAPPL Class 265
GAUTHHOST Class 268
GFILE Class 271
GDEPLOYMENT Class 275
GHNODE Class 280
GHOST Class 283
GPOLICY Class 286
GROUP Class 291
GSUDO Class 296
GTERMINAL Class 299
GWINSERVICE Class 302
HNODE Class 306
HOLIDAY Class 314
HOST Class 318

HOSTNET Class 320
HOSTNP Class 324
KMODULE Class 326
LOGINAPPL Class 330
MFTERMINAL Class 337
POLICY Class 341
PROCESS Class 346
PROGRAM Class 350
PWPOLICY Class 357
REGKEY Class 358
REGVAL Class 362
RESOURCE_DESC Class 366
RESPONSE_TAB Class 367
RULESET Class 368
SECFILE Class 373
SECLABEL Class 376
SEOS Class 377
SPECIALPGM Class 382
SUDO Class 387
SURROGATE Class 392
TCP Class 396
TERMINAL Class 401
UACC Class 406
USER Class 409
USER_ATTR Class 419
USER_DIR Class 421
WEBSERVICE Class 423
WINSERVICE Class 424
XGROUP Class 428
XUSER Class 432
Classes in the Windows Environment 442
COM Class 442
DEVICE Class 443
DISK Class 445
DOMAIN Class 447
FILE Class 448
GROUP Class 451
OU Class 452
PRINTER Class 453
PROCESS Class 454
REGKEY Class 455
REGVAL Class 457
SEOS Class 458

SERVICE Class 460
SESSION Class 462
SHARE Class 463
USER Class 466
Classes in the UNIX Environment 472
FILE Class 472
GROUP Class 472
USER Class 473
Classes for Custom Purposes 473
User Defined Class 473
Unicenter TNG User-Defined Class 473

Appendix A: Windows Values 475

Windows File Attributes 475
Windows Account Flags 476
Windows Permissions 477
Windows Privileges 478

Chapter 1: Introduction

This section contains the following topics:

About this Guide (see page 15)
Who Should Use this Guide (see page 15)

About this Guide

This guide provides information about CA Access Control selang command, database classes and properties, and Windows values. This guide is also provided with CA Access Control Premium 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 selang commands or maintaining and configuring a CA Access Control-protected environment.

Chapter 2: The selang Command Language

This section contains the following topics:

CA Access Control Command Line Interpreter (see page 17)
selang Command Syntax (see page 25)
selang Command Authorization (see page 26)
selang Environments (see page 33)
selang Configuration on UNIX (see page 34)
Get selang Help (see page 36)

CA Access Control Command Line Interpreter

CA Access Control is administered through a command shell known as selang, the CA Access Control command language. The selang command language lets you make definitions in the CA Access Control database. The selang command language is the command definition language.

The selang utility is located in the bin directory of your CA Access Control installation. When you enter the selang shell, a special selang prompt appears. The exact form of the prompt depends on your working environment. It looks similar to this:

```
AC>
```

The selang command shell operates on the local database by default. To operate on the CA Access Control database of a different station, specify the hosts command before entering the selang commands.

More information:

selang Environments (see page 33)
hosts Command—Connect to a Remote CA Access Control Terminal (see page 119)

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 the superuser 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.

More information:

hosts Command—Connect to a Remote CA Access Control Terminal (see page 119)

Features of the selang Command Shell

After you enter the selang command shell, the following prompt appears:

```
AC>
```

When the prompt appears, you can enter selang commands. Enter commands separated with a semi-colon (;). If you need to enter a command on more than one line, type a backslash (\) at the end of a line to continue typing the command on the next line. You can edit the command line. Use the left and right arrow keys to move around within the line. You can insert characters by typing them directly in place, and delete characters with the standard Backspace and Delete keys, or on UNIX, by pressing Ctrl+D.

selang supports many of the command line entry features available in UNIX shell tcsh and other smart shells. These include the following features:

- Special characters
- Shortcut keys
- Command history
- Special features

Note: On UNIX, you can use a *UNIX exit* which is a program that you can specify-a shell script or an executable-to run automatically before or after a user or group is added or updated. For more information about UNIX exits, see the *Endpoint Administration Guide for UNIX*.

Special Characters

selang supports the following special characters:

Character	Description	Meaning
# or *	Pound (hash) or asterisk	At the beginning of a line, indicates that the line is a comment; the line is not executed. Comment lines are useful when inputting the selang commands from a file.
!	Exclamation mark	At the beginning of the line, indicates that the rest of the line is a shell command. selang sends the command to the operating system shell program for execution; CA Access Control does not execute the line.
\	Backslash	As the last character of a line, indicates the command continues on the following line.
;	Semicolon	Terminates a command and introduces a new command on the same line.
|	Pipe	Sends the output of the preceding command to the input of the succeeding command (the specified *pipe*).

Shortcut Keys

selang supports the following shortcut keys:

Key	Valid on	Meaning
Up-arrow, Down-arrow, or ^	All	Used to navigate and retrieve a command from the command history.
Tab	UNIX	Serves for word completion.

Key	Valid on	Meaning
Ctrl+D	UNIX	With the cursor positioned at the end of the line, displays a list of words that match the word completion string in the command line. With the cursor positioned anywhere else on the line, deletes the character to the right of the cursor.
Esc, Esc Ctrl+2	UNIX	Displays the help text for the command in the command line. All the text in the command line is preserved, so that you can continue typing the command from where you left off.
F1	Windows	Inserts the previous command, character by character.
F2	Windows	Displays a window with the instruction: "Enter char to copy up to:" When you enter a character from the previous command, selang enters the command up to the first instance of the character. If the character occurs more than once in the command, you can press F2 again to insert up to the next instance. Use Backspace to cancel.
F3	Windows	Enters the previous command (same as up arrow).
F4	Windows	Edits the previous instruction. Displays a window with the instruction: "Enter char to delete up to:" Use Backspace to cancel.
F5	Windows	Enters the previous command (same as up arrow).
F6	Windows	Enters a Ctrl Z (^Z) in the command line. This allows you to press Enter and continue entering the command on the next line.
F7	Windows	Displays a window listing the command history. You can use the up and down arrows to select any previous command. Use Esc to cancel.
F8	Windows	Enters the previous command, as the up arrow does, but with the cursor positioned at the beginning of the command line rather than at the end.
F9	Windows	Displays a window with the instruction: "Enter command number:" The number you enter inserts the command with the corresponding number in the F7 listing. Use Esc to cancel.

Command History

selang stores executed commands in a *history list*. Use the up and down arrow keys to display commands in the command line from the history list. To see only the commands that start with a specific character or string, type the beginning of the command before using the up and down arrows. When you press *Enter*, the text currently displayed in the command line is executed.

To view previously issued commands, enter the history command.

The selang command shell supports the following shortcuts that use the commands stored in the history list:

Shortcut	Runs
^^ [*string*]	The previous command. If you specify *string*, selang appends it to the original command.
^*n* [*string*]	The *n*th command in the history list, where *n* is a positive integer. If you specify *string*, selang appends it to the original command.
^-*n* [*string*]	The *n*th command from the end of the list, where *n* is a positive integer. If you specify *string*, selang appends it to the original command.
^*mask* [*string*]	The most recently issued command that begins with *mask*, where *mask* is a text string. If you specify *string*, selang appends it to the original command.

Note: On Windows, you can use the F7 key to view the history list.

Special Features

You can use several additional techniques to save keystrokes in the selang command shell.

Note: Record and class names are case-sensitive on UNIX but not on Windows.

- Command Recognition

  selang recognizes which command you want to execute as soon as you have typed in enough characters to distinguish it from all the other available commands. For example, you can type **ho** to run the *hosts* command as it is the only command beginning with those letters. As soon as you type **ho**, selang can recognize the intended command. On the other hand, there are several commands that begin with the string **new**. You must add enough characters to distinguish between newusr, newgrp, newfile, and newres.

- Abbreviations

  Each command is also associated with a one- to four-letter abbreviation. For example, because there are several commands beginning with the string new, you can also use the abbreviation **nu** for the command **newusr**. These abbreviations are documented as part of the command syntax for each command. You can enter commands in either uppercase or lowercase.

- Word Completion (UNIX only)

  Press *Tab* in the middle of a word to complete the word. Word completion is context-sensitive. If more than one word matches the specified string, selang uses the shortest word or word fragment that matches the string. For example, if you type the letter *n*, selang supplies ew, to form the word new. If this is not the required word, type another one or two characters and press *Tab* again to complete the word. Press Ctrl+D to see all the possible options. This is useful if you are not sure which command to use. Using the example in the previous paragraph, if you add the letter u to the word new and press *Tab*, selang supplies sr, giving you the command newusr.

  Words that are not part of the selang commands are stored in memory for use by the word completion feature later on in the same session. For example, if you type newusr Mercedes, and then later type showusr Me followed by *Tab*, the abbreviation Me is expanded to Mercedes, as follows:

  showusr Mercedes

  This assumes that you have not entered any other user name that begins with "Me".

Wildcard Matching

selang supports the following wildcard characters:

*** (asterisk)**

Any sequence of zero or more characters.

? (question mark)

Any single character (except a path separator for files).

To make a single character a "do not care" character that matches any other single character, use a question mark (?), as in the following examples:

Specify this...	To do this...
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 (*), as in the following examples:

Specify this...	To do this...
i.c	main.c, list.c
st*.h	stdio.h, stdlib.h, string.h
*	All records of the specified class

selang Command Syntax

Each selang command performs a specific action on the CA Access Control database. The syntax of a selang command is:

commandname *parameters*

The command name tells CA Access Control which command to execute. You usually follow the command with one or more parameters that supply CA Access Control with additional information needed to execute the command.

The syntax of a selang parameter is:

parameterName[(arguments)]

The parameter name identifies the parameter to CA Access Control. Many parameters require arguments that provide CA Access Control with the information necessary to process the parameter. Some parameters accept more than one argument. When more than one argument is specified, separate the arguments with a comma or a space. The argument of a parameter may itself be a parameter.

To remove a record property when a string defines the argument, simply enter the property with empty parenthesis **()**. In some cases, you can use an asterisk (*****) as an argument, to cover all possible values for that argument. If you use an asterisk, the asterisk does *not* override earlier or later commands that give specific values to the same argument. Moreover, if the argument is a file name, you can use a wildcard as part of a file name pattern. The wildcards are * (for zero or more characters) and ? (for one character).

In the UNIX environment, user-supplied information is case-sensitive and can consist of both lowercase and uppercase letters. For example, you may specify the full name of the user whose user ID is user53 as Mike Jones. Windows does not recognize case-sensitive information *but still saves it*. If you administer a remote Windows host from a UNIX workstation, UNIX will look for user-supplied information *as stored*. For example, if a user is identified in a Windows environment as Mike Jones, you may enter his name as mike jones when administering the local CA Access Control database. However, if you want to administer the database from a remote UNIX machine, you must enter his name as Mike Jones.

selang Command Authorization

To use selang commands that change records in the AC or native operating system (native OS) environment, you must have sufficient authority. For most commands, *one* of the following conditions must be met:

- You are the owner of the resource.
- You have the ADMIN attribute.
- The resource record is within the scope of a group in which you have the GROUP-ADMIN attribute.
- You have CREATE or MODIFY access authority in the ACL of the record in the ADMIN class.
- (Windows) If your installation only permits management of the native Windows environment, you are a member of the CA Access Control Administrators group in the Windows database.
- (UNIX) If your installation only permits management of the native UNIX environment, you are a member of the CA Access Control Administrators group in the security files of the local UNIX host.

Note: Exceptions to these general rules are noted in the description of each command.

Access Control List Support

To give or deny access authority, you can use seven types of access control lists:

ACL

Standard access control list that contains the user names and group names authorized to access the resource and the level of access granted to each.

NACL

Negative access control list that contains the user names or group names that are not authorized to access the resource.

PACL

Program access control list that depends upon the accessing program. Each PACL contains the user names and group names, the level of access, and the name of the program or shell script the user must execute to access the particular resource.

INET-ACL

Internet access control list.

CACL

Conditional access control list.

CALACL

Calendar access control, a resource ACL that depends upon the Unicenter TNG calendar.

AZNACL

The authorization ACL; an ACL that allows access to a resource based on the resource description.

CA Access Control uses all relevant lists when it checks a user's authority to access a resource.

Note: You can maintain any single list with a single authorize command. To change more than one list you need to issue authorize again. You cannot define multiple access rights for multiple users and groups with one authorization rule. You must separate the rules.

The following table lists which access control lists you can use with each class. Classes that do not appear in the table have no access control lists and cannot be controlled by the authorize command.

Class	ACL/ NACL	CALACL	PACL	INET-ACL	CACL	AZNACL
ADMIN	X	X	X			
APPL	X	X				X
AUTHHOST	X	X				X
CONNECT	X	X	X			
CONTAINER	X	X	X			
DOMAIN	X	X	X			
FILE	X	X	X			
GAPPL	X	X				X
GAUTHHOST	X	X				X

Class	ACL/ NACL	CALACL	PACL	INET-ACL	CACL	AZNACL
GFILE	X	X	X			
GHOST				X		
GSUDO	X	X				
GTERMINAL	X	X				
HOLIDAY	X	X				
HOST				X		
HOSTNET				X		
HOSTNP				X		
LOGINAPPL	X	X				
MFTERMINAL	X	X	X			
PROCESS	X	X	X			
PROGRAM	X	X				
REGKEY	X	X	X			
REGVAL	X	X	X			
SUDO	X	X	X			
SURROGATE	X	X	X			
TCP	X	X	X		X	
TERMINAL	X	X	X			
UACC	X	X				
USER_DIR	X					X

Access Authority by Class

Valid access values depend on the class the resource belongs to. The following table lists valid access values by class in the AC environment.

Class	Valid Access Values	Lets Accessors...
All classes	all	Perform *all* valid operations for the class.
	none	Perform *no* valid operations for the class.
ADMIN	create	Create records in this class.
	delete	Delete records in this class.

Class	Valid Access Values	Lets Accessors...
	join	Add a group to a USER record and to complete the linking of a user to a group. **Note:** The accessor must also have *modify* access.
	modify	Modify existing records. **Note:** To link a user to a group (add user names to GROUP records) the accessor must also have *join* access.
	password	Change the passwords of other users. **Note:** This access type affects only the USER class.
	read	List records in this classes
AUTHHOST	read	Login from an authenticated host.
CONNECT	read	Connect to the remote host.
CONTAINER	*inherited*	**Note:** Valid access values for this class are the valid values for the class of the contained objects.
DOMAIN	chmod	Create and delete trust relationships between one domain and another. **Note:** Both domains must have this access type.
	execute	Add or delete members from the domain.
	read	List domain members.
FILE, GFILE	chdir	Access the directory with the equivalent of read and execute permissions.
	chmod	Change file system modes. **Note:** Only applicable on UNIX hosts.
	chown	Change the owner of the record.
	control	Perform *all* valid operations except *delete* and *rename*.
	create	Create records in this class.
	delete	Delete records in this class.
	execute	Execute a program. **Note:** The accessor must also have *read* access.
	read	Use a file or directory without changing it. **Note:** On UNIX, if you want *read* privileges to control whether users can perform operations that obtain information about the file (such as ls -l), set the STAT_intercept configuration setting to 1. For more

Class	Valid Access Values	Lets Accessors...
		information, see the *Reference Guide*.
	rename	Rename to a record in this class.
	sec	Change the ACL of records in this class.
	update	Perform the combined operations of *read*, *write*, and *execute*.
	utime	Change the modification time of a file. **Note:** Only applicable on UNIX hosts.
	write	Change the file or directory.
HNODE	read	List records in the class.
	write	Edit the details of the record.
HOLIDAY	read	Log in during the specified holiday.
KMODULE	load	Load a kernel module.
	unload	Unload a kernel module.
MFTERMINAL	read	Log in from the Mainframe terminal.
	write	Administer from the Mainframe terminal.
POLICY	delete	Delete the policy.
	execute	Deploy the policy.
	read	View policy details.
	write	Edit the details of the record.
	undeploy	Perform the combined operations of *delete* and *execute*.
PROCESS	read	Kill the process.
PROGRAM, SUDO, GSUDO	execute	Execute a program.
REGKEY	delete	Delete a Windows registry key.
	read	List the contents of the Windows registry key.
	write	Change the Windows registry key.
REGVAL	delete	Delete a Windows registry value.
	read	Read a Windows registry value.
	write	Change a Windows registry value.
RULESET	read	View the details of the record.

Class	Valid Access Values	Lets Accessors...
	write	Edit the details of the record.
SURROGATE	execute	Surrogate to the user.
TCP	read	Access TCP services from remote hosts or host groups.
TERMINAL, GTERMINAL	read	Log in to the terminal.
	write	Administer the terminal.
UACC	*inherited*	**Note:** Valid access values for this class are the valid values for the class it is defining.
WINSERVICE	read	View the properties of the Windows service.
	start	Start the Windows service.
	modify	Change the properties of the Windows service.
	resume	Resume a paused Windows service.
	stop	Stop a Windows service.
	pause	Pause a Windows service.

Note: The values none and all are applicable to all classes. The value all represents the entire group of access values, other than none, for a particular class. For more information about access authority, see the *Endpoint Administration Guide* for your OS.

Windows Access Authority by Class

Valid access values depend on the class the resource belongs to. The following table lists valid access values by class in the Windows (nt) environment.

Class	Valid Access Values	Let Accessors...
All classes	all	Perform *all* valid operations for the class.
	none	Perform *no* valid operations for the class.
COM, DISK	change	Perform the combined operations of *delete*, *read*, and *write*.
	changepermissions	Modify the ACL of the resource.
	delete	Delete the resource.
	read	Access data on the resource without changing it.

Class	Valid Access Values	Let Accessors...
	takeownership, chown, owner	Change the owner of the specified resource.
	write	Write data to the specified resource.
FILE		**Note:** It is only possible to define access authorities for NTFS files; FAT files cannot have access authorities.
	change	Perform the combined operations of *delete*, *read*, and *write*.
	changepermissions, sec	Modify the ACL of the resource.
	chmod	Perform all operations except *delete*.
	chown	Change the owner of the specified resource.
	delete	Delete the resource.
	execute	Execute programs. **Note:** To use this access, the accessor must also have *read* access.
	read	Access a resource without changing it.
	rename	Renames the resource. **Note:** To rename a file, you must have *delete* access to the source and *rename* access to the target. The audit log reflects this order of events.
	write	Modify the resource.
	update	Perform the combined operations of *read*, *write*, and execute.
PRINTER	manage	Manage the printer. For example, set the data for a specified printer, pause printing, resume printing, clear all print jobs, update the ACL, or change printer properties.
	print	Print using the printer.
REGKEY	append, create, subkey	Create or modify a subkey of the registry key
	takeownership, chown, owner	Change the owner of the resource
	changepermissions, sec, dac, writedac	Modify the ACL of the resource.
	delete	Delete the resource.
	enum	Enumerate subkeys.
	link	Create a link to the registry key.

Class	Valid Access Values	Let Accessors...
	notify	Change notifications for a registry key or for subkeys of a registry key.
	query	Query a value of the registry key
	read	Access a resource without changing it.
	readcontrol, manage	Read the information in the registry key's security descriptor, not including the information in the system (audit) ACL.
	set	Create or set a value of the registry key.
	write	Change the registry key and its subkeys.
SHARE	change	Change properties of the resource or remove sharing from the resource.
	read	Access a resource without changing it.

Note: The values *none* and *all* are applicable to all classes. The value *all* represents the entire group of access values, other than *none*, for a particular class. For more information about access authority, see the *Endpoint Administration Guide for Windows*.

selang Environments

In addition to working on the local CA Access Control database, selang can be used to modify the native (Windows or UNIX) database, the local Policy Model database (PMDB), a database on a remote host (Windows or UNIX) where CA Access Control is installed, or on CA Access Control configuration settings. To switch environments, use the *env* (environment) command, which is available in all environments.

Some commands are the same in the different environments, but they may have different parameters and arguments. You should, therefore, check the syntax carefully when beginning to work in a new environment.

Note: When you are entering the *native* property of a command using env, the command is entered in both the *native* environment and current environment.

The following environments are supported:

Environment	Command	Prompt	Description
Policy Model	env pmd	AC(pmd)>	All selang commands operate on the local PMDB.
Native Windows	env nt	AC(nt)>	All selang commands modify the Windows database.
AC	env ac	AC>	All selang commands operate on the CA Access Control database. **Note:** This is the default.
Native UNIX	env unix	AC(unix)>	All selang commands operate on the security files of the local UNIX host.
Native	env native	AC(native)>	All selang commands operate in the native environment of the host.
Remote Configuration	env config	AC(config)>	All selang commands operate on the CA Access Control configuration settings for the host.

More information:

environment Command—Set the Security Environment (see page 112)
Classes in the AC Environment (see page 220)
Classes in the UNIX Environment (see page 472)
Classes in the Windows Environment (see page 442)

selang Configuration on UNIX

On UNIX, you can manage the way selang works. Most of the options are concerned with the way selang manages the UNIX security system (for the selang UNIX environment).

The selang utility uses the following two files for configuration options:

seos.ini

Contains CA Access Control configuration options. This is the main configuration file for CA Access Control.

lang.ini

Contains configuration information that selang uses.

selang uses the lang.ini files in *one or both* of the following locations:

- The directory where the seos.ini file is located.
- The user's home directory.

If you specify a token in only one of these lang.ini files, selang uses the value from that file. If you specify a token differently in the two lang.ini files, the value in the user's home directory overrides the other one.

The values for the tokens DefaultShell and DefaultHome in the server's seos.ini file *override* the values set in the tokens DefaultShell and HomeDirPrefix in the lang.ini file.

Note: Sample lang.ini files are located in the directory *ACInstallDir*/samples/lang.init.

Change the User File

The default file for updating UNIX users is /etc/passwd but you can change this default. This is normally required on the NIS server if you are working under NIS.

To change the user file modify the *YpServerPasswd* in the *passwd* section of the seos.ini file to point to your user's file full pathname.

Change the File for Updating Groups

The default file for updating UNIX groups is /etc/group but you can change this default. This is normally required on the NIS server if you are working under NIS.

To change the file for updating groups modify the **YpServerGroup** in the **passwd** section of the **seos.ini** file to point to your user's file full pathname.

Automatic Backup of the UNIX User and Group Files

Before the first update of a UNIX user in a session and before the first update of a UNIX group in a session, CA Access Control creates a backup copy of the files /etc/passwd or /etc/group. The backup files are called /etc/passwd.SeOS.bak and /etc/group.SeOS.bak, respectively. If an error occurs when updating the UNIX system, the original information is recoverable. Backups are made only before the first change to the UNIX system in a selang command shell session.

Get selang Help

You can get help at any time in the interactive selang command environment.

To enter selang online help, enter one of the following:

? or help

The selang online help text for the environment you are in, appears on the screen, displaying the table of contents.

help topic

topic

Defines a selang command or other topic related to the selang command shell.

The help text that describes the topic appears.

help env

env

Defines is a selang environment.

The help text for the specified environment appears on the screen, displaying the table of contents.

Note: On UNIX, to display the help text for a command typed in the command line without deleting the text in the command line, type Ctrl+2 (or press Esc, Esc).

More information:

help Command—Get selang Help (see page 118)
selang Environments (see page 33)
selang Commands Reference (see page 37)

Chapter 3: selang Commands

This section contains the following topics:

selang Commands Reference (see page 37)
selang Commands in the AC Environment (see page 42)
selang Commands in the Remote Configuration Environment (see page 158)
selang Commands in the Native UNIX Environment (see page 163)
selang Commands in the Native Windows Environment (see page 176)
selang Commands in the Policy Model Environment (see page 208)

selang Commands Reference

The following table lists all selang commands alphabetically.

Note: Commands that operate in the same manner in all environments are documented only in the AC environment. However, some commands are valid in more than one environment but differ in the manner they operate in each environment. These commands are marked with an asterisk (*) in the *Description* column of the table below and are documented separately in each environment they are valid in.

Command	Short	Environments	Description
alias		AC and unix **Note:** For UNIX hosts only.	Lists or defines aliases for selang commands and properties.
authorize	auth	AC and nt	*Sets the authority a specific accessor has when accessing a specific resource.
authorize-	auth-	AC and nt	*Removes the authority previously given to a specific accessor when accessing a specific resource.
backuppmd		pmd	Backs up the data in the PMDB database to a specified directory.
check		AC	Checks whether a user has access privileges to a particular resource.
checklogin		AC	Determines a user's login privileges, whether a password check is needed, and whether a terminal access check is needed.

Command	Short	Environments	Description
checkpwd		AC	Checks a user's new password, without changing it, to make sure it follows password rules.
chfile	cf	AC and native	*Changes the definition of a file record in the CA Access Control or native OS database.
chgrp	cg	AC and native	*Changes existing internal group settings in the CA Access Control or native OS database.
chres	cr	AC and nt	*Changes an existing resource record in the CA Access Control or native OS database.
chusr	cu	AC and native	*Changes an existing internal user in the CA Access Control or native OS database.
chxgrp	cxg	AC	Changes existing enterprise group settings in the CA Access Control database.
chxusr	cxu	AC	Changes existing enterprise user settings in the CA Access Control database.
createpmd		pmd	Creates a PMDB on a remote host.
deletepmd		pmd	Removes the PMDB's selang protection files, the contents of the PMDB directory, and the PMDB directory from the remote host.
deploy		AC	Executes deployment selang commands stored in a RULESET object for the particular POLICY.
deploy-		AC	Executes policy undeployment selang commands stored in a RULESET object for the particular POLICY.
editfile	ef	AC and native	*Adds or changes the definition of a file record in the CA Access Control or native OS database.
editgrp	eg	AC and native	*Adds a new group to, or changes existing group settings in, the CA Access Control or native OS database.
editres	er	AC and nt	*Adds a new resource record to, or changes an existing resource record in, the CA Access Control or native OS database.
editres config		config	Lists the configuration settings in the source you specify.
editusr	eu	AC and native	*Adds a new user to, or changes an existing user in, the CA Access Control or native OS database.

Command	Short	Environments	Description
editxgrp	exg	AC	Adds a new enterprise group or changes existing enterprise group properties, in the CA Access Control database.
editxusr	exu	AC	Adds a new enterprise user or changes existing enterprise user properties, in the CA Access Control database.
end_transaction		AC	Completes the start_transaction command for Dual Control PMDB processes.
environment	env	*all*	Sets the security environment selang is operating on.
find	f	AC and native	Lists the classes in the environment or the records in a class.
findpmd		pmd	Lists all PMDBs on the computer.
find config		config	Lists sources of configuration settings (ini files or registry entries) you can manage on this host.
find file		native	Lists system files.
find xgroup		nt	Lists the names of enterprise groups in the current or trusted domains.
find xuser		nt	Lists the names of enterprise users in the current or trusted domains.
get dbexport		AC	Retrieves the rules that were exported from a CA Access Control or PMD database.
get devcalc		AC	Retrieves policy deviation calculation results.
help		*all*	Displays selang help.
history		*all*	Displays the commands issued previously in the session.
hosts		*all*	Shows or sets the host to which selang commands are sent.
join	j	AC and native	*Joins a user to a group.
join-	j-	AC and native	*Removes a user from a group.
joinx	jx	AC	Joins an enterprise user to a group
joinx-	jx-	AC	Removes an enterprise user from a group.
list		AC and native	An alias of the *find* command.
listpmd		pmd	Lists information about the PMDB and its subscribers, update file, and error log.

Command	Short	Environments	Description
newfile	nf	AC	Adds the definition of a file record in the CA Access Control database.
newgrp	ng	AC and native	*Adds a new group to the CA Access Control or native OS database.
newres	nr	AC and nt	*Adds a new resource record to the CA Access Control or native OS database.
newusr	nu	AC and native	*Adds a new internal user to the CA Access Control or native OS database.
newxgrp	nxg	AC	Adds a new enterprise group to the CA Access Control database.
newxusr	nxu	AC	Adds a new enterprise user to the CA Access Control database.
pmd		pmd	Clears the Policy Model error log, updates the subscriber list, releases subscribers, starts and stops the Policy Model service, truncates the update file, and reloads the initialization files.
rename		AC	Renames an object in the database.
restorepmd		pmd	Restores a PMDB on a local host.
rmfile	rf	AC	Removes a file resource record from the CA Access Control database.
rmgrp	rg	AC and native	*Removes a group from the CA Access Control or native OS database.
rmres	rr	AC and nt	*Removes a resource record from the CA Access Control or native Windows database.
rmusr	ru	AC and native	*Removes a user from the CA Access Control or native OS database.
rmxgrp	rxg	AC	Removes an enterprise group from the CA Access Control database.
rmxusr	rxu	AC	Removes an enterprise user from CA Access Control
ruler		AC and native	Sets the properties that display when a show command is executed.
search		AC and native	An alias of the *find* command.
setoptions	so	AC and nt	*Sets or displays the global options that control the behavior of the database.

Command	Short	Environments	Description
showfile	sf	AC and native	*Lists the properties of file records in the CA Access Control or native OS database.
showgrp	sg	AC and native	*Lists the properties of group records in the CA Access Control or native OS database.
showres	sr	AC and nt	*Lists the properties of records in the CA Access Control or native Windows database.
showres config		config	Lists the configuration settings in the source you specify.
showusr	su	AC and native	*Lists the properties of user records in the CA Access Control database or the native OS database.
showxusr	sxu	AC	Lists the properties of enterprise user records in CA Access Control.
source		*all*	Executes the commands in a specified file.
start dbexport		AC	Exports a CA Access Control or PMD database.
start devcalc		AC	Triggers a policy deviation calculation.
start_transaction		AC	Starts recording a file that contains an unprocessed transaction for Dual Control PMDB processes, with one or more commands.
subs		pmd	Adds a subscriber to a parent PMDB or subscribes a database to a parent PMDB.
subspmd		pmd	Changes the parent of the database in the host to which you are connected.
unalias		AC and unix	Removes aliases for selang commands and properties.
undeploy		AC	An alias of the deploy- command.
unsubs		pmd	Removes subscribers from the subscriber list of a PMDB.
xaudit		nt	Sets auditing criteria and begins logging access events.
xaudit-		nt	Removes auditing criteria and stops logging access events.

Note: The native environment conforms to the rules of either the Windows (nt) or UNIX environments, depending on the operating system of the host to which you are connected.

selang Commands in the AC Environment

This section contains a complete alphabetic reference to all the selang commands that operate on the CA Access Control database (commands in the AC environment).

alias Command—Define selang Aliases

Valid on UNIX hosts

Use the alias command to list or define aliases for selang commands and properties. Any user can use the alias command.

Note: You can build a set of aliases to use in all selang sessions by defining those aliases in a startup file and using the *selang -r* command.

This command has the following format:

alias [*aliasName* [*aliasValue*]]

aliasName

(Optional) Defines the name you want to use as alias.

If this option is not specified, the alias command lists all defined aliases.

aliasValue

(Optional) Defines the meaning that the selang command shell should associate with *aliasName*.

If this option is not specified, the alias command displays the value of the specified alias.

You can also include up to ten variables in *aliasValue* ($0 to $9). If *aliasValue* contains variables, you must replace each variable with the proper value in parentheses when invoking the alias.

Example: Use a Variable to Ease the Creation of New Administrators

To create an alias that makes adding new administrators to the database easier, enter the following command:

```
alias newadm newusr ($0) admin
```

To use this alias, simply add the names of the new administrators in brackets. For example:

```
newadm(Terri)
```

This adds a user called Terri to the database. Terri is given the ADMIN attribute which is required for administering the database. This is the same as entering the following command:

```
newusr Terri admin
```

Example: Simplify Property Names

To create an alias that replaces the property name *access* with the shortened alias *acc*, enter the following command:

```
alias acc access
```

You can now enter the following to use this alias:

```
authorize file x uid(y) acc(z)
```

Example: Use Aliases in Context

Aliases are not simply expanded variables; they are only interpreted in a context where a command name or a property name should be specified. For example, define the alias:

```
alias newterm newres terminal
```

Then issue the following command:

```
newterm newterm owner(nobody)
```

The first newterm string is replaced but not the second as the context requires the second instance of the string to be a terminal name. This is the same as entering the following command:

```
newres terminal newterm owner(nobody)
```

More information:

unalias Command—Remove selang Aliases (see page 157)
selang Utility—Run the CA Access Control Command Line (see page 18)

authorize Command—Set Access Authorities on a Resource

Valid in the AC environment

Use the authorize command to change accessors' access authorities to a resource.

This command modifies an access control list associated with a resource. It changes only one entry in an access control list at a time.

When an accessor attempts to access a resource, CA Access Control checks the appropriate access control lists to determine the access authority. These access control lists include those that are in the resource record, and can also include access control lists in resource group records. If an accessor is denied access authority in any NACL that covers the resource, the authority is denied, even if the authority is granted in another ACL.

The owner of a resource always has all access authorities to the resource. If you want to change the access authority of the user who is the owner, change the resource to have a different owner, for example, the user nobody.

Note: This command also exists in the Windows environment, but operates differently there.

To use the authorize command, you need sufficient authority, which means that one or more of the following must be true:

- You have the ADMIN attribute.
- You have the GROUP-ADMIN attribute for a resource group of which the resource is a member.
- You are the owner of the resource.
- You have modify access authority in the ADMIN class record that corresponds to the resource.

The authorize command has different forms for different sets of classes. These sets are:

- TCP
- HOST, GHOST, HOSTNET, and HOSTNP
- All other classes

This command has the following format for the TCP class:

{authorize|auth} TCP *tcpServiceName* \
[{access|deniedaccess}(*accessType*)] \
[ghost(*ghostName [,ghostName]...*)] \
[host(*hostName [,hostName]...*)] \
[hostnet(*hostNetName [,hostNetName]...*)] \
[hostnp(*hostNamePattern [,hostNamePattern]...*)] \
{gid|uid|xgid|xuid}(*accessor* [,*accessor*]...]) ...

This command has the following format for the HOST, GHOST, HOSTNET, and HOSTNP classes:

{authorize|auth} {HOST|GHOST|HOSTNET|HOSTNP} *stationName*
[{access|deniedaccess}(*accessType*)] \
service({*serviceName*|*serviceNumber*|*serviceNumberRange*}) \
{ gid | uid | xgid | xuid}(*accessor* [,*accessor*...]) ...

This command has the following format for all other classes:

{authorize|auth} *className resourceName* \
[{access|deniedaccess}(*accessType*)] \
[calendar(*calendarName*)] \
[{unix|nt}]\
[via (pgm (*program* [,*program*]...))] \
{ gid | uid | xgid | xuid}(*accessor* [,*accessor*...]) ...

access (*accessType*)

Defines the access authority entry in the resource ACL access control list. This ACL specifies which access authorities are granted to accessors.

accessType

Defines the access type in the resource ACL, for example, read or write.

Note: If you omit both the access(*accessType*) and the deniedaccess(*accessType*) options to the authorize command, CA Access Control assigns the access that is specified by the implicit access property of the record in the UACC class for the class of resource (for example in the UACC file record if the resource is a file).

calendar(*calendarName*)

Specifies the calendar to use for determining access authority.

className

Defines the class to which *resourceName* belongs.

deniedaccess(*accessType)*

Changes the access authority in the resource NACL. The NACL specifies which access types are denied to accessors.

accessType

Specifies the access type to be denied, for example, read, or write.

gid (*accessor* [*,accessor...*])

Defines one or more internal groups for whom you want to set the access authority.

ghost(*ghostName [,ghostName]...*)

Defines one or more group hosts for which you want to set access authority to the TCP/IP service.

host(*hostName [,hostName]...*)

Defines one or more hosts for which you want to set access authority to the TCP/IP service.

hostnet(*hostNetName [,hostNetName]...*)

Defines one or more HOSTNET records for which you want to set access authority to the TCP/IP service.

hostnp(*hostNamePattern [,hostNamePattern]...*)

Defines one or more HOSTNP records for which you want to set access authority to the TCP/IP service.

nt

Specifies whether to add values to the system ACLs in Windows.

Valid for the FILE class only.

resourceName

Defines the resource record whose access control list is being modified.

service(*serviceName*|*serviceNumber*|*serviceNumberRange*)

Defines the services the local host is permitted to provide to the remote host or hosts.

serviceNumber |*serviceNumberRange*

Defines the service number or range.

Specify a range as two integers separated by a -(hyphen), for example, 1-99.

Limits: An integer in the range 0 to 65535.

stationName

Specifies the record name within the indicated class, as follows:

- **HOST—**Name of single station.
- **GHOST—**Name of a group of hosts as defined in the database by the ghost command.
- **HOSTNET—**Name of a group of hosts as defined by a set of mask and match values for the IP address.
- **HOSTNP—**Name of a group of hosts as defined by a name pattern.

For hosts that cannot be resolved, specify the IP address range in IPv4 format.

tcpServiceName

Specifies the CA Access Control TCP service record whose access authority you are setting.

uid (*accessor* [*,accessor...*])

Defines one or more internal users for whom you want to set the access authority.

You can use * to represent all internal users.

unix

Specifies whether to add values to the system ACLs in UNIX.

Valid only on UNIX environments that support ACLs, and only for records in the FILE class.

via(pgm(*programName [,programName]...*))

Defines one or more programs for conditional program access. The via parameter specifies an entry in the PACL of the resource. *programName* specifies a program that can access the resource. *programName* can contain wildcard characters. If a program matches several entries in a PACL, the entry with the longest non-wildcard match takes precedence.

If *programName* specifies a program or shell script that is not defined in the PROGRAM class, CA Access Control automatically creates a PROGRAM record to protect it.

xgid (*accessor* [*,accessor...*])

Defines one or more enterprise groups for whom you want to set the access authority.

xuid (*accessor* [*,accessor...*])

Defines one or more enterprise users for whom you want to set the access authority.

Example: Authorize Angela to Read a File

The following selang command authorizes enterprise user Angela to read the file protected by the FILE resource /projects/secrets:

```
auth FILE /projects/secrets xuid(Angela) access(read)
```

Example: Authorize Only Angela to Read a File

The following selang commands authorize enterprise user Angela, but nobody else, to read the file protected by the FILE resource /projects/secrets:

```
auth FILE /projects/secrets xuid(Angela) access(read)
auth FILE /projects/secrets defaccess (none)
chres FILE /projects/secrets owner(nobody)
```

Note: On UNIX, if you want *read* privileges to control whether users can perform operations that obtain information about the file (such as ls -l), set the STAT_intercept configuration setting to 1. For more information, see the *Reference Guide*.

Example: Authorize All Users in a Group to Log in to a Terminal

The following selang command authorizes all members of the enterprise group RESEARCH to log in to the terminal protected by the TERMINAL resource *tty10*:

```
auth TERMINAL tty10 xgid(RESEARCH) access(read)
```

Example: Authorize Joe to Back up Files

The following selang command authorizes enterprise user Joe to back up the files protected by the GFILE resource secret_files:

```
auth GFILE secret_files xuid(Joe) \
via(pgm(/bin/backup)) access(read)
```

For a Windows endpoint, an equivalent command is as follows:

```
auth GFILE secret_files xuid(Joe) \
via(pgm(C:\WINDOWS\system32\ntbackup.exe)) access(read)
```

These commands only have an effect if the Joe's access authority is not determined by the ACL or NACL of the resource.

More information:

authorize- Command—Remove Access Authorities from a Resource (see page 49)
authorize Command—Set Accessors' Authority to Access Windows Resources (see page 176)
chfile Command—Modify File Records (see page 56)
ch[x]grp Command—Change Group Properties (see page 62)
chres Command—Modify Resource Records (see page 74)
ch[x]usr Command—Change User Properties (see page 90)
authorize- Command—Remove Accessors' Authority to Access Windows Resources (see page 178)

authorize- Command—Remove Access Authorities from a Resource

Valid in the AC environment

Use the authorize- command to remove accessors from the access control lists (ACLs) of a resource.

Note: This command also exists in the native Windows environment but operates differently there.

You need the same access authority to use the authorize- command as you do to use the authorize command.

The authorize- command has different formats for different sets of classes. These sets are:

- TCP
- HOST, GHOST, HOSTNET, and HOSTNP
- All other classes

This command has the following format for the TCP class:

{authorize-|auth-} TCP *tcpServiceName* \
{gid |uid |xgid |xuid } (*accessorName* [,*accessorName*]...)\
[host(*hostName [,hostName]*...)] \
[ghost(*ghostName [,ghostname]*...)] \
[hostnet(*hostNetName [,hostNetName]*...)] \
[hostnp(*hostNamePattern [,hostNamePattern]*...)]

This command has the following format for the HOST, GHOST, HOSTNET, and HOSTNP classes:

{authorize-|auth-} *className stationName* \
service({*serviceName* | *serviceNumber* |*serviceNumberRange*})

This command has the following format for all remaining classes:

{authorize-|auth-} *className resourceName* \
[{access-|deniedaccess-}]\
[calendar(*calendarName*)] \
{gid |uid |xgid |xuid } (*accessorName* [,*accessorName*]...)

access-

Specifies that the command should remove accessors from the resource ACL (which grants access authorities), rather than from the NACL.

If neither access- or deniedaccess- are specified, the command removes the accessors from both ACLs.

calendar(*calendarName*)

Removes the calendar specified for determining access authority.

className

Specifies the name of the class to which *resourceName* belongs.

deniedaccess-

Specifies that the command should remove accessors from the resource NACL (which denies access authority), rather than from the ACL.

gid (*accessor* [,*accessor*]...)

Defines one or more internal groups whose entries are to be removed. Separate each *accessor* with a comma or space.

ghost(*ghostName*)

Specifies the name of an object in class GHOST.

host(*hostName*)

Specifies the name of an object in class HOST.

hostnet(*hostNetName*)

Specifies the name of an object in class HOSTNET.

hostnp(*hostNamePattern*)

Specifies a pattern defined in class HOSTNP.

nt

Specifies whether to remove values from the system ACLs in Windows.

Valid for the FILE class only.

resourceName

Specifies the name of the resource record whose access control list is being modified. Specify only one resource record.

service(*serviceName*|*serviceNumber*|*serviceNumberRange*)

Defines the services you want to remove from an ACL.

stationName

Specifies the record name within the indicated class, as follows:

- **HOST—**Name of single station.
- **GHOST—**Name of a group of hosts as defined in the database by the ghost command.
- **HOSTNET—**Name of a group of hosts as defined by a set of mask and match values for the IP address.
- **HOSTNP—**Name of a group of hosts as defined by a name pattern.

For hosts that cannot be resolved, specify the IP address range.

serviceNumber |*serviceNumberRange*

Defines the service number or range.

Specify the range as two integers separated by a -(hyphen), for example, 1-99.

Limits: An integer in the range 0 to 65535

uid (*accessor* [,*accessor*]...)

Defines one or more internal users whose entries are to be removed. Separate each *accessor* with a comma or space.

You can use uid(*) to specify all internal users.

unix

Specifies whether to remove add from the system ACLs in UNIX.

Valid only on UNIX environments that support ACLs, and only for records in the FILE class.

xgid (*accessor* [,*accessor*]...)

Defines one or more enterprise users whose entries are to be removed. Separate each *accessorName* with a comma or space.

xuid (*accessor* [,*accessor*]...)

Defines one or more enterprise groups whose entries are to be removed. Separate each *accessor* with a comma or space.

Example: Remove a group authority to access a file

The following command removes the group research from both the ACL and NACL of the file covered by the resource /products/new:

```
auth- FILE /products/new xgid(research)
```

The research group now has the default access to the file.

More information:

authorize Command—Set Access Authorities on a Resource (see page 44)
authorize Command—Set Accessors' Authority to Access Windows Resources (see page 176)
authorize- Command—Remove Accessors' Authority to Access Windows Resources (see page 178)
chfile Command—Modify File Records (see page 56)
ch[x]grp Command—Change Group Properties (see page 62)
chres Command—Modify Resource Records (see page 74)
ch[x]usr Command—Change User Properties (see page 90)

check Command—Determine a User's Access Authority

Valid in the AC environment

Use the check command to determine if a user has access privileges to a particular resource. The command checks access according to the resource's ACL and default access property. However, it does not support PACLs; that is, it does not indicate whether the user can access a resource using a specific program.

Note: This command is not available when seos is down. For more information about PACLs, see the *Endpoint Administration Guide* for your OS.

To use this command you must have sufficient authority over the resource, as defined by any of the following conditions:

- The process running the command has the SERVER attribute.
- You have the ADMIN attribute.

This command has the following format:

```
check className resourceName uid(userName) access(authority)
```

access(*authority*)

Defines the access authority to be checked for the accessor identified by the uid parameter.

Valid values depend on the resource being checked.

className

Defines the name of the class to which *resourceName* belongs.

resourceName

Defines the name of the resource record.

uid(*userName*)

Defines the name of the CA Access Control user whose authority to access *resourceName* is to be verified.

Example: Determine whether a user has access to a resource

To determine whether user Alain has write access to the resource *testfile* of class *file*, enter the following command:

```
check FILE /testfile uid(Alain) access(w)
```

The following sample output of this command indicates that user Alain has write access to the defined file because Alain is the resource's owner:

```
Access to FILE /testfile GRANTED
Stage: Resource OWNER check
```

checklogin Command—Determine Login Information

Valid in the AC environment

Use the checklogin command to determine a user's login privileges, whether a password check is needed, and whether a terminal access check is needed.

Note: This command is not available when seos is down.

To use this command you must have sufficient authority over the resource, as defined by any of the following conditions:

- The process running the command has the SERVER attribute.
- You have the ADMIN attribute.

This command has the following format:

```
checklogin userName [password(password)] [terminal(terminalName)]
```

password(*password*)

(Optional) Defines the password that CA Access Control checks against the operating system password and the database, if password checking is enabled.

userName

Defines the name of the user whose right to login is being verified.

terminal(*terminalName*)

(Optional) Defines the terminal that CA Access Control checks to determine if a user has login privileges from it.

Example: Determine whether user has login privileges

To determine whether user Frank has login privileges to the *localhost* from terminal *mutra*, enter the following command:

```
checklogin Frank terminal(mutra)
```

The following output of the command, indicates that user Frank can login from terminal mutra to host winsome (localhost):

```
Login by USER frank to host winsome is GRANTED
Stage: Resource class global universal access
```

To verify user Frank's password, enter the following command:

```
checklogin frank password(111) terminal(localhost)
```

To verify user Frank's password against the one in the CA Access Control database, enter the following commands:

```
so class+(PASSWORD) (localhost)
checklogin frank password(moonshine) terminal(tack)
```

The *so* command above enables password checking.

checkpwd Command—Check a Password for Compliance

Valid in the AC environment

Use the checkpwd command to check a user's password for compliance with password rules. This check does not change the password.

To use this command you must be a superuser with the ADMIN attribute.

A new password is accepted or rejected according to CA Access Control password rules:

- If a new password is accepted, the following success message displays:

  Changing *userName*'s password is permitted.

- If a new password is rejected, the following fail message displays:

  Changing *userName*'s password is denied.
  denied_reason

  Where *denied_reason* is the actual password rule that did not pass.

  For example:

  Changing *JDoe*'s password is denied.
  Too few lowercase letters in password.

Only the first rule that the password fails appears in the *denied_reason*. If, for example, a password is too short, *and* the password has too few capital letters, only *Password is too short* appears.

Note: This command is not available when seos is down. For more information about password rules, see the *Endpoint Administration Guide* for your OS.

This command has the following format:

checkpwd *userName* password(*newPassword*)

userName

Specifies the name of the CA Access Control user whose new password you want to check.

password(*newPassword)*

Specifies the password you want to check.

chfile Command—Modify File Records

Valid in the AC environment

Use the chfile, editfile, and newfile commands to work with records in the FILE class. These commands are identical in structure and only vary in the following way:

- The chfile command *modifies* one or more records in the FILE class.
- The editfile command *creates or modifies* one or more records in the FILE class.
- The newfile command *creates* one or more records in the FILE class.

Note: This command also exists in the native environment but operates differently.

To add or change a record for a file belonging to the FILE class, you must have sufficient authority over the file. CA Access Control makes the following checks until one of the following conditions is met:

1. You have the ADMIN attribute.
2. The resource record is within the scope of a group in which you have the GROUP-ADMIN attribute.
3. When changing a record, that you are its owner.
4. You have CREATE (for newfile or editfile) or MODIFY (for chfile) access authority in the ACL of the FILE record in the ADMIN class.
5. That you are the owner of the file (when defining a file to CA Access Control that exists in the native OS), if the token use_unix_file_owner in the seos.ini file is set to yes.

```
{{chfile|cf}|{editfile|ef}|{newfile|nf}} filename... \
      [audit{none|all|success|failure}] \
      [category[-](categoryName)] \
      [comment(string)|comment-] \
      [defaccess(accessAuthority)] \
      [label(labelName)|label-] \
      [level(number)|level-] \
      [notify(mailAddress)|notify-] \
      [gowner(groupName)] \
      [owner({userName|groupName})] \
      [restrictions( \
            [days({anyday|weekdays|{[mon] [tue] [wed] \
                  [thu] [fri] [sat] [sun]}})] \
            [time({anytime|startTime:endTime}) \
      |restrictions-] \
      [warning|warning-]
```

audit{none|all|success|failure}

Specifies which access events are logged. The types of access are:

- **all**-CA Access Control logs both authorized accesses and detected unauthorized access attempts.
- **failure**-CA Access Control logs detected unauthorized access attempts. This is the default value.
- **none**-CA Access Control does not write any records in the log file.
- **success**-CA Access Control logs authorized accesses to the resource.

Note: To use the audit parameter, you must have the AUDITOR attribute.

category(*categoryName*)

Defines a space- or comma-separated list of security category records (defined in the CATEGORY class) to assign to the file.

If you specify the category parameter when the CATEGORY class is not active, CA Access Control updates the definition of the file in the database; however, the updated category assignment has no effect until the CATEGORY class is activated again.

Note: For more information about security category checking, see the *Endpoint Administration Guide* for your OS.

category-(*categoryName*)

Deletes one or more security categories from the resource record. When removing more than one security category, separate the security category names with a space or a comma.

The specified security categories are deleted from the resource record, regardless of whether the CATEGORY class is active.

Note: This parameter is only valid when modifying a record.

comment(*string*)

Adds an alphanumeric string of up to 255 characters to the file record. If the string contains any blanks, enclose the string in single quotation marks. The string replaces any existing comment defined previously.

comment-

Deletes the comment string from the file record.

Note: This parameter is only valid when modifying a record.

defaccess(*accessAuthority*)

Specifies the default access authority for the file. The default access authority is the authority granted to any accessor that requests access to the file, but that is not in the access control lists of the file. The default access is also applied to users who are not defined in the database.

fileName

Defines the name of the file record. At least one file name must be specified.

If you are adding or changing a record in class FILE using a generic file name, use the wildcard expressions permitted in selang. When defining or changing more than one record, enclose the list of file names in parentheses and separate the file names with a space or a comma.

Note: If more than one file name is specified, CA Access Control processes each file record independently in accordance with the specified parameters. If an error occurs while processing a file, CA Access Control issues a message and continues processing with the next file in the list.

gowner(*groupName*)

Assigns a CA Access Control group as the owner of the file record. The group owner of the file record has unrestricted access to the file, provided the group owner's security level, security label, and security category authorities are sufficient to allow access to the file. The group owner of the file is always permitted to update and delete the file record.

label(*labelName*)

Assigns to the file a security label defined in the SECLABEL class. A security label represents an association between a particular security level and zero or more security categories. If the resource record currently contains a security label, the security label specified here replaces the current security label.

Note: For more information about security label checking, see the *Endpoint Administration Guide* for your OS.

label-

Deletes the security label defined in the file record.

Note: This parameter is only valid when modifying a record.

level(*number*)

Assigns a security level to the resource record. Enter a positive integer between 1 and 255. If a security level was previously assigned to the resource record, the new value replaces the existing value.

Note: For more information about security level checking, see the *Endpoint Administration Guide* for your OS.

level-

Stops CA Access Control from performing security level checking for the resource.

Note: This parameter is only valid when modifying a record.

notify(*mailAddress*)

Instructs CA Access Control to send notification messages whenever the file represented by the resource record is successfully accessed. Enter a user name, an email address of a user, or the email address of a mail group if an alias is specified.

Notification takes place only when the Log Routing System is active. The notification messages are sent either to the screen or to the mailbox of the users, depending on the setup of the Log Routing System.

Each time a notification message is sent, an audit record is written in the audit log.

The recipient of notify messages should log in frequently to respond to the unauthorized access attempts described in each message.

Limit: 30 characters.

Note: For information about filtering and viewing audit records, see the *Endpoint Administration Guide* for your OS.

notify-

Specifies that no one is notified when CA Access Control grants access to the file represented by the record.

Note: This parameter is only valid when modifying a record.

owner(*Name*)

Assigns a CA Access Control user or group as the owner of the file record. The owner of the file record has unrestricted access to the file, provided the owner's security level, security label, and security category authorities are sufficient to allow access to the file. The owner of the file is always permitted to update and delete the file record.

restrictions(days(*dayData*) time(*timeData*))

Specifies the days of the week and the hours in the day when the file is accessible to users.

If you omit the days argument and specify the time argument, the time restriction applies to any day-of-week restriction already indicated in the record. If you omit time and specify days, the day restriction applies to any time restriction already indicated in the record. If you specify both days and time, the users are allowed to access the system only during the specified time period on the specified days.

days(***dayData***)

Specifies the days on which users can access the file. The days argument takes the following sub-arguments:

- **anyday**-Gives access to the file on any day.
- **weekdays**-Gives access to the resource only on weekdays-Monday through Friday.
- **mon tue wed thu fri sat sun**-Gives access to the resource only on the specified days. You can specify the days in any order. If more than one day is specified, separate the days with a space or a comma.

time(*timeData*)

Specifies the period during which users can access the file. The time argument takes the following sub-arguments:

- **anytime-**Gives access to the resource at any time of the day.
- *startTime*:*endTime*-Gives access to the resource only during the specified period. The format of both *startTime* and *endTime* is *hhmm*, where *hh* is the hour in 24-hour notation (00 through 23) and *mm* is the minutes (00 through 59). Note that 2400 is not a valid time value. *StartTime* must be less than *endTime*, and both times must occur on the same day. If the terminal is in a different time zone from the processor, adjust the time values by translating the start and end times for the terminal to the equivalent local times for the processor. For example, if the processor is in New York and the terminal is in Los Angeles, to allow access to the terminal from 8:00 a.m. to 5:00 p.m. in Los Angeles, specify time(1100:2000).

restrictions-

Deletes any restrictions that limit the ability to access the file.

Note: This parameter is only valid when modifying a record.

warning

Puts the file into Warning mode.

warning-

Takes the file out of Warning mode.

Example: Restrict Access to a File to All but the Superuser

To restrict access to the /etc/passwd file to READ access to all users except the superuser, enter the following command:

```
chfile /etc/passwd defaccess(read) owner(root)
```

The following must be true:

- You have the ADMIN attribute.
- The record /etc/passwd is defined in the database.
- There are no entries in the ACL of the record /etc/passwd.

Example: Restrict Access to a File by Time

To prevent access to the /home/bob/secrets file and let the owner access the file only on weekdays between 08:00 and 18:00, enter the following command:

```
newfile /home/bob/secrets defac(none) restrictions(d(weekdays) t(0800:1800))
```

The following must be true:

- You have the ADMIN attribute.
- Bob is a CA Access Control user and is the owner of the /home/ bob/secrets record in the FILE class.

Example: Prevent Access to Your Home Directory

To prevent all other users from accessing any file in your home directory **(**/home/bob), enter the following command on UNIX:

```
newfile /home/bob/* defaccess(none)
```

You can do the same on Windows using the following command:

```
newfile %userprofile%\* defaccess(none)
```

The following must be true:

- You are defined to CA Access Control.
- You are the native owner of the file.

More information:

authorize Command—Set Access Authorities on a Resource (see page 44)
rmfile Command—Delete File Records (see page 128)
showfile Command—Display File Properties (see page 143)
chfile Command—Modify UNIX File Settings (see page 163)
chfile Command—Modify Windows File Settings (see page 180)
Access Authority by Class (see page 28)

ch[x]grp Command—Change Group Properties

Valid in the AC environment

Use the commands chgrp, chxgrp, editgrp, editxgrp, newgrp, and newxgrp to change the properties of groups, and to create the groups in the CA Access Control database if necessary.

These commands all have synonyms, as follows:

- chgrp—cg
- chxgrp—cxg
- editgrp—eg
- editxgrp—exg
- newgrp—ng
- newxgrp—nxg

These commands are identical in structure, and vary only in their scope, in the following ways:

- The chgrp, editgrp, and newgrp commands work with records in the GROUP class. These let you create or modify CA Access Control groups without reference to the enterprise user store. The differences between these commands are as follows:
  - The chgrp command *modifies* one or more records in the GROUP class.
  - The editgrp command *creates or modifies* one or more records in the GROUP class.
  - The newgrp command *creates* one or more records in the GROUP class.

  Note: These commands also exist in the native environment but operate differently there.

- The chxgrp, editxgrp and newxgrp commands work on records in the XGROUP class. These let you create or modify CA Access Control groups that are defined in the enterprise user store. The differences between them are as follows:
  - The chxgrp command *modifies* one or more records in the XGROUP class.
  - The editxgrp command *creates or modifies* one or more records in the XGROUP class.
  - The newxgrp command *creates* one or more records in the XGROUP class.

Authorization Required

To create a new CA Access Control group, at least one of the following conditions must be true:

- You have the ADMIN attribute.
- You are assigned the CREATE authority in the access control list of the GROUP or XGROUP record in the ADMIN class.

To add or modify a group, at least one of the following conditions must be true:

- You have the ADMIN attribute.
- The group record is within the scope of a group in which you have the GROUP-ADMIN attribute.
- You are the owner of the group.
- You are assigned the MODIFY (for ch[x]grp) or CREATE (for edit[x]grp) authority in the access control list of the GROUP or XGROUP record in the ADMIN class.

{{chgrp|cg}|{chxgrp|cxg}|{editgrp|eg}|{editxgrp|exg}|{newgrp|ng}|{newxgrp|nxg}} *groupName* ...
[audit(none|all|success|failure|loginsuccess|loginfail|trace)|audit-] \
[comment(*string*)|comment-] \
[expire[(*mm/dd/yy*[*yy*[*@hh:mm*]])]|expire-] \
[gowner(*groupName*)] \
[homedir(*fullPath*|nohomedir)] \
[inactive(*numInactiveDays*)|inactive-] \
[maxlogins(*maximumNumberOfLogins*)|maxlogins-] \
[mem(*groupName*)|mem+(*groupName*)|mem-(*groupName*)]\
[name('*fullName*')] \
[nt[(comment(*comment*))]
[owner(*userName*|*groupName*)] \
[parent(*groupName*)|parent-] \

```
[password( \
        [history(numberStoredPasswords)|history-] \
        [interval(maximumPasswordChangeInterval)|interval-] \
        [min_life(minimumPasswordChangeInterval)|min_life-] \
        [rules( \
                [alpha(minimumAlphaCharacters)] \
                [alphanum(minimumAlphanumericCharacters)] \
                [bidirectional|bidirectional-] \
                [grace(numberOfGraceLogins)] \
                [min_len(minimumPasswordLength)] \
                [max_len(maximumPasswordLength)] \
                [lowercase(minimumLowercaseCharacters)] \
                [max_rep(maxRepetitiveCharacters)] \
                [namechk|namechk-] \
                [numeric(minimumNumericCharacters)] \
                [oldpwchk|oldpwchk-] \
                [special(minimumSpecialCharacters)] \
                [uppercase(minimumUppercaseCharacters)] \
                [use_dbdict|use_dbdict-] \
        )|rules-] \
        )] \
[pmdb(PolicyModelName)|pmdb-] \
[restrictions( \
        [days({anyday|weekdays|{[mon] [tue] [wed] \
                [thu] [fri] [sat] [sun]}})] \
        [time(anytime|startTime:endTime) \
|restrictions-] \
[resume[(mm/dd/yy[yy][@hh:mm])]|resume-] \
[shellprog(fullPath)] \
[supgroup(superiorGroup)|supgroup-] \
[suspend[(mm/dd/yy[yy][@hh:mm])]|suspend-] \
[unix[( \
        [appl(quotedString)] \
        [groupid(groupidNumber)] \
        [userlist(userName...)] \
)]] \
```

To remove any record property where the property is defined by a string, type the property followed immediately by either - (minus sign), or () (empty parenthesis).

Note: Some parameters are relevant only when a group functions as a profile group. A profile group cannot be an enterprise group.

audit(*mode*)

Turns on the trace audit for this command. The audit modes are: none, all, success, failure, loginsuccess, loginfail, trace.

audit-

Turns off the trace audit for this command.

comment(*string*)

Adds to the group record a comment string of up to 255 alphanumeric characters (single-byte). If the string contains spaces, enclose the entire string in single quotation marks. The string replaces any existing string that you added previously.

Note: In German, only 128 characters are recorded.

comment-

Deletes the comment string, if any, from the group record. Use this parameter only with the chgrp or editgrp command.

expire(*date*)

Sets the date on which the accounts of the group members expire. If you do not specify a date, the user accounts expire immediately, provided the users are not currently logged in. If the users are logged in, the accounts expire when the users log out. This parameter applies only to profile groups.

Specify the expiration date, and optional time, in the following format: *mm/dd/yy [yy][@HH:MM].* Year can be either 2 or 4 digits.

Note: You cannot enable expired user records by specifying the resume parameter with a resume date. Use the expire- parameter to enable expired user records.

expire-

For the newgrp command, defines user accounts that do not have an expiration date. For the chgrp and editgrp commands, removes the expiration date from the user accounts. This parameter applies only to profile groups.

gowner(*groupName*)

Assigns a CA Access Control user or group as the owner of the group record. When you specify more than one group name, enclose the names in parentheses and separate the group names with a space or a comma. If you add a group to the database and omit this parameter, you are the owner of the group record.

grace(*numberOfGraceLogins*)

Sets the maximum number of logins that are permitted before the users are suspended. The number of grace logins must be between 0 and 255. After the number of grace logins is reached, the users are denied access to the system and must contact the system administrator to select a new password. If grace is set to zero, the users cannot log in. This parameter applies only to profile groups.

grace-

Deletes the grace login setting for the group. Use this parameter only with the chgrp or editgrp command. This parameter applies only to profile groups.

groupName

Specifies the name of the group you are creating or whose properties you are changing. For the command new[x]grp, each group name must be unique and must not currently exist in the database. However, a group and a user can share the same name.

history

Specifies the number of stored passwords. You can eliminate the history file with history-.

homedir(*fullPath*|nohomedir)

Specifies the full path of the users' home directories. If the path you specify ends with a slash, *groupName* is concatenated to the specified path. If you specify nohomedir then a home directory is not automatically set.

inactive(*numInactiveDays*)

Specifies the number of days that must pass before the system changes users to inactive status. When the number of days is reached, users cannot log in. This parameter applies only to profile groups.

Enter a positive integer or zero for *numInactiveDays*. If inactive is set to zero, the effect is the same as using the inactive- parameter.

Note: In the user record, inactive users are not marked. To identify inactive users, you must compare the Last Accessed Time value with the Inactive Days value.

inactive-

Changes the users' status from inactive to active. Use this parameter only with the chgrp or editgrp command. This parameter applies only to profile groups.

interval(*maximumPasswordChangeInterval*)

Sets the number of days that must pass after the password was set or changed before the system prompts the user for a new password. Enter a positive integer or zero. An interval of zero disables password interval checking for the group so that the password does not expire. The default set by the setoptions command is not used. Set an interval of zero only for users with low security requirements.

When the specified number of days is reached, CA Access Control informs the user that the current password has expired. The user can immediately renew the password or continue using the old password until the number of grace logins is reached. After the number of grace logins is reached, the user is denied access to the system and must contact the system administrator to select a new password. This parameter applies only to profile groups.

interval-

Cancels the password interval setting for the group. If canceled, any value in the user record is used. Otherwise, the default set by the setoptions command is used. Enter this parameter only with the chgrp or editgrp command. This parameter applies only to profile groups.

maxlogins(*maximumNumberOfLogins*)

Sets the maximum number of terminals users can log in to at the same time. A value of 0 (zero) means that users can log in from any number of terminals concurrently. If this parameter is not specified, any value in the user record is used. Otherwise, the global maximum logins setting is used. This parameter applies only to profile groups.

Note: If maxlogins is set to 1, you cannot run selang. You must shut down CA Access Control, change the maxlogins setting to greater than one, and start CA Access Control again.

maxlogins-

Deletes the group's maximum login setting. If this parameter is not specified, any value in the user record is used. Otherwise, the global maximum logins setting is used. Use this parameter only with the chgrp or editgrp command. This parameter applies only to profile groups.

mem(*GroupName*) | mem+(*GroupName*)

Adds members groups (or child groups) to the group in CA Access Control. The member groups *(GroupName*) must already be defined in CA Access Control. If you are adding more than one member group, separate the group names with a comma. If a group name contains a space, enclose it in quotation marks.

Note: To add users to a internal group, use the join[x] command.

This option applies to internal groups only.

mem-(*GroupName*)

Removes member groups from this group. The member groups (*GroupName*) must already be defined in CA Access Control. If you are removing more than one member group, separate the group names with a comma. If a group name contains a space, enclose it in quotation marks.

Note: To remove users from a internal group, use the join[x]- command.

This option applies to internal groups only.

min_life(*minimumPasswordChangeInterval*)

The minimum number of days that must pass before users are allowed to change the password again. This parameter applies only to profile groups.

min_life-

Deletes the min_life setting of a group. If this parameter is not specified and the min_life parameter is set in a user record, the value in the user record is used. Otherwise, the global min_life setting is used. Use this parameter only with the chgrp or editgrp command. This parameter applies only to profile groups.

name(*fullname*)

Specifies the full name of the group. Enter an alphanumeric string of up to 47 characters. If the string contains any blanks, enclose the string in single quotation marks.

nt(*nt-group-attributes*)

(Windows only) Adds or changes the group definition in the local Windows system.

comment('*comment*')

Adds a comment string to the native record. If you previously added a comment string to the record, the new string specified here replaces the existing string.

comment is an alphanumeric string of up to 255 characters. If the string contains any blanks, enclose the entire string in single quotation marks.

owner(*Name*)

Assigns a CA Access Control user or group as the owner of the group record. If you are adding a group to the database and you omit this parameter, you are the owner. See the *Endpoint Administration Guide* for your OS for more information.

parent(*groupName*)

Assigns an existing CA Access Control group as the parent group of the group record. See the *Endpoint Administration Guide* for your OS for more information on parent and child relationships.

parent-

Deletes the link between a group and its parent group. Use this parameter only with the chgrp or editgrp command.

password

Assigns a password to this group.

rules

Specifies rules for the password:

alpha(*minimumAlphaCharacters*)

Minimum number of alphabetic characters.

alphanum(*minimumAlphanumericCharacters*)

Minimum number of characters.

bidirectional|bidirectional-

Specifies whether to use bidirectional password encryption. If bidirectional password encryption is enabled, each new password is encrypted and can be decrypted back to clear text. This encryption gives a wider comparison between new passwords and old passwords (password history). When bidirectional encryption is disabled, one-way password history encryption is activated, and you cannot decrypt old passwords.

Note: You must set history to a value greater than 1 to use this feature.

Note: On UNIX, you must also set the configuration setting passwd_format to NT to use this feature.

min_len(*minimumPasswordLength*)

Minimum password length.

max_len(*maximumPasswordLength*)

Maximum password length.

lowercase(*minimumLowercaseCharacters*)

Minimum number of lowercase characters.

max_rep(*maximumRepetitiveCharacters*)

Maximum number of repeated characters.

namechk|namechk-

Check password against name.

numeric(*minimumNumericCharacters*)

Minimum number of numeric characters.

oldpwchk|oldpwchk-

Check password against old password.

special(*minimumSpecialCharacters*)

Minimum number of special characters.

uppercase(*minimumUppercaseCharacters*)

Minimum number of uppercase characters.

use_dbdict|use_dbdict-

Sets the password dictionary. use_dbdict sets the token to **db** and compares passwords against words in the CA Access Control database. use_dbdict- sets the token to **file** and checks passwords against a file specified in the seos.ini file for UNIX or Windows registry for Windows.

password-

Deletes the need for a password for this group.

pmdb(*PolicyModelName*)

Specifies that when a user in the group changes a password with the utility sepass, the new password is propagated to the specified Policy Model. Enter the fully qualified name of the PMDB.

The password is not sent to the Policy Model defined in the parent_pmd or passwd_pmd token in the [seos] section of seos.ini. This parameter applies only to profile groups.

pmdb-

Removes the PMDB attribute from the group record. Use this parameter only with the chgrp or editgrp command. This parameter applies only to profile groups.

restrictions(days(*dayData*) time(*timeData*))

Specifies the days of the week and the hours in the day when members of the group are allowed to log in to the system.

CA Access Control does not force a user off the system if the login period expires while the user is logged in. Also, the login restrictions do not apply to batch jobs; a user can run a background process at any time. This parameter applies only to profile groups.

If you omit the days argument and specify the time argument, the time restriction applies to any day-of-week restriction already indicated in the record. If you omit time and specify days, the day restriction applies to any time restriction already indicated in the record. If you specify both days and time, the members of the group are allowed to access the system only during the specified time period on the specified days.

days(***dayData***)

Specifies the days on which users can log in to the system. The days argument takes the following sub-arguments:

- **anyday**-Lets users log in on any day.
- **weekdays**-Lets users log in only on weekdays-Monday through Friday.
- **mon tue wed thu fri sat sun**-Lets users log in only on the specified days. You can specify the days in any order. If more than one day is specified, separate the days with a space or a comma.

time(***timeData***)

Specifies the period during which users can log in to the system. The time argument takes the following sub-arguments:

- **anytime-**Lets users log in at any time of the day.
- *startTime***:***endTime*-Lets users log in only during the specified period. The format of both *startTime* and *endTime* is *hhmm*, where *hh* is the hour in 24-hour notation (00 through 23) and *mm* is the minutes (00 through 59). Note that 2400 is not a valid time value. If *endTime* is a smaller number than *endTime*, the period is considered to extend across midnight. Otherwise, it is considered to take place on a single day.

  Note: CA Access Control uses the time zone of the processor. If the user logs in at a terminal in a different time zone from the processor, you must take this into account.

restrictions-

Deletes any restrictions that limit the users' ability to log in to the system from the group record. If this parameter is not specified and the restrictions parameter is set in a user record, the value in the user record is used. Use this parameter only with the chgrp or editgrp command. This parameter applies only to profile groups.

resume(*date*)

Enables user records that were disabled by specifying the suspend parameter. Enter a date, and optional time, in the following format: *mm/dd/yy[@HH:MM].*

If you specify both the suspend parameter and the resume parameter, the resume date must fall after the suspend date. If you omit *date*, the user is enabled immediately on execution of the chgrp command. See the *Endpoint Administration Guide* for your OS for more information. This parameter applies only to profile groups.

resume-

Erases the resume date, and time if used, from the group record. Consequently, the status of the users is changed from active (enabled) to suspended. Use this parameter only with the chgrp or editgrp command. This parameter applies only to profile groups.

shellprog(*fullPath*)

Specifies the full path of the initial program or shell that is executed after the user invokes the login or su command. *FullPath* is a character string.

supgroup(*Group'sSuperiorGroup*)

Specifies a supergroup (or parent group).

suspend(*date*)

Disables user records, but leaves them defined in the database. Enter a date, and optional time, in the following format: *mm/dd/yy[@HH:MM].*

A user cannot use a suspended user account to log in to the system. If *date* is specified, the user records are suspended on the specified date. If *date* is omitted, the user records are suspended immediately upon execution of the chgrp command. This parameter applies only to profile groups.

suspend-

Erases the suspend date from the user records, changing the status of the users from disabled to active (enabled). Use this parameter only with the chgrp or editgrp command. This parameter applies only to profile groups.

unix(*groupidNumber*)

(UNIX only) Sets group attributes on UNIX or creates the group if it does not already exist.

The *groupidNumber* is a decimal number. You cannot specify a group ID of zero. If you omit the number, CA Access Control finds the largest current group ID and sets the ID of the group to this number plus one. CA Access Control creates group ID numbers in the same way when adding or modifying more than one group at a time. The token AllowedGidRange in the seos.ini file may define certain unavailable numbers.

userlist*(userName)*

Assigns members to the group. *UserName* is the user name of one or more UNIX users. When assigning more than one user, separate the user names with a comma or a space. For the chgrp and editgrp commands, the member list specified here replaces any member list that is currently defined for the group.

Examples

- The user Bob wants to change the parent group and owning group for the enterprise group Sales from ACCOUNTS to PAYROLL.

  chxgrp Sales parent(PAYROLL) owner(PAYROLL)

- The user Admin1 wants to change the parent of group projectB from divisionA to divisionB and assign the group RESEARCH as the new owner.

  Admin1 has the ADMIN attribute.

  chxgrp projectB parent(divisionB) owner(RESEARCH)

- The admin user Sally wants to remove the home directory and the shell program specifications for the group profile NewEmployee.

  Sally is the owner of NewEmployee.

  editgrp NewEmployee homedir() shellprog()

- The user Admin1 wants to add the group ProjectA as a child group of the group RESEARCH. The user Admin1 is to be the owner of the ProjectA group.

  Admin1 has the ADMIN attribute.

  The default is owner(Admin1).

  newgrp ProjectA parent(RESEARCH)

More information:

join[x] Command—Add Users to Internal Groups (see page 122)
join[x]- Command—Remove Users from Groups (see page 125)
rm[x]grp Command—Delete Group Records (see page 129)
show[x]grp Command—Display Group Properties (see page 145)
chgrp Command—Modify UNIX Groups (see page 165)
chgrp Command—Modify Windows Groups (see page 181)

chres Command—Modify Resource Records

Valid in the AC environment

Use the chres, editres, and newres commands to work with resource records that belong to a CA Access Control class. These commands are identical in structure and only vary in the following way:

- The chres command *modifies* one or more resources.
- The editres command *creates or modifies* one or more resources.
- The newres command *creates* one or more resources.

Note: This command also exists in the native Windows environment but operates differently there.

To add a resource using the newres command, at least one of the following conditions must be true:

- You have the ADMIN attribute.
- You have CREATE access authority in the ACL of the resource class's record in the ADMIN class.
- If the token use_unix_file_owner in the seos.ini file is set to yes, an owner of a file in UNIX can define it as a new resource to CA Access Control.

To add or change a resource using the chres or editres commands, you must have sufficient authority over the resource. CA Access Control checks in the following order for any *one* of these conditions:

1. You have the ADMIN attribute.
2. The resource record is within the scope of a group in which you have the GROUP-ADMIN attribute.
3. You are the owner of the record.
4. You are assigned MODIFY (for chres) or CREATE (for editres) access authority in the access control list of the resource class's record in the ADMIN class.

Note: The maximum length of a resource name is 255 characters.

The following table lists command parameters that apply for each class that can be administered using the chres, editres, and newres commands.

Class	Properties											
	audit	calendar	category	comment	defaccess	label	level	notify	owner	restrictions[-]	warning	other
ACVAR				X					X			VARIABLE_ TYPE, VARIABLE_ VALUE
ADMIN	X	X	X	X	X	X	X	X	X	X	X	
CALENDAR				X					X			
CATEGORY				X					X			
CONNECT	X	X	X	X	X	X	X	X	X	X	X	
CONTAINER	X	X		X					X		X	MEM
DOMAIN	X	X	X	X	X	X	X	X	X	X	X	MEM
FILE	X	X	X	X	X	X	X	X	X	X	X	
GFILE	X	X		X				X	X		X	MEM
GHOST	X	X		X					X	X	X	MEM
GSUDO		X		X	X				X			MEM
GTERMINAL	X	X		X	X				X	X		MEM
HNODE	X	X	X	X	X	X	X	X	X	X	X	SUBSCRIBER, POLICY
HOLIDAY	X		X	X	X	X	X	X	X	X	X	DATES
HOST	X	X		X					X	X	X	
HOSTNET	X	X		X					X		X	MASK, MATCH
HOSTNP	X	X		X					X	X	X	

Class	Properties											
	audit	**calendar**	**category**	**comment**	**defaccess**	**label**	**level**	**notify**	**owner**	**restrictions[-]**	**warning**	**other**
LOGINAPPL	X	X		X	X			X	X	X	X	LOGINFLAGS, LOGINMETHOD, LOGINPATH, LOGINSEQUENCE
MFTERMINAL	X	X	X	X		X	X	X	X		X	DAYTIME
POLICY	X	X	X	X	X	X	X	X	X	X	X	SIGNATURE, RULESET
PROCESS	X	X	X	X	X	X	X	X	X	X	X	
PROGRAM	X	X	X	X	X	X	X	X	X	X	X	TRUST
PWPOLICY				X					X			
REGKEY	X	X		X	X			X	X		X	DAYTIME
REGVAL	X	X		X	X			X	X		X	DAYTIME
RULESET	X	X	X	X	X	X	X	X	X	X	X	SIGNATURE, CMD, UNDOCMD
SECFILE				X					X			TRUST, FLAGS
SECLABEL			X	X			X		X			
SEOS		X	X	X		X	X					HOST
SPECIALPGM				X					X			
SUDO	X	X	X	X	X	X	X	X	X	X	X	TARGUID,

Class	Properties											
	audit	**calendar**	**category**	**comment**	**defaccess**	**label**	**level**	**notify**	**owner**	**restrictions[-]**	**warning**	**other**
												PASSWORD
SURROGATE	X	X	X	X	X	X	X	X	X	X	X	
TCP	X		X	X	X	X	X	X	X	X	X	
TERMINAL	X	X	X	X	X	X	X	X	X	X	X	
UACC	X		X	X	X				X			
USER-ATTR									X		X	
USER-DIR	X			X					X			

{{chres|cr}|{editres|er}|{newres|nr}} *className resourceName* \
[ac_id(*id*)] \
[audit({none|all|success|failure})] \
[calendar[-](*calendarName*)] \
[category[-](*categoryName*)] \
[cmd+(*selang_command_string*)|cmd-] \
[comment(*string*)|comment-] \
[container[-](*containerName*)] \
[dates(*time-period*)] \
[dh_dr{-|+}(*dh_dr*)] \
[disable|disable-] \
[defaccess(*accessAuthority*)] \
[filepath(*filePaths*)] \
[flags[-|+](*flagName*)] \
[gacc(*access-value*)] \
[gowner(*groupName*)] \
[host(host-name)|host-] \
[label(*labelName*)|label-] \
[level(*number*)|level-] \
[mask(*inetAddress*)|match(*inetAddress*)] \
[mem(*resourceName*)|mem-(*resourceName*)] \
[node_alias{-|+}(*alias*)] \
[node_ip{-|+}(*ip*)] \
[notify(*mailAddress*)|notify-] \
[of_class(*className*)] \
[owner({*userName* | *groupName*})] \
[{password | password-}] \
[policy(name(policy-name) {{deviation+|dev+}|{deviation-|dev-}})] \
[policy(name(policy-name) status(policy-status) {updator|updated_by}(user-name))] \
[{restrictions([days({anyday|weekdays|{[mon] [tue] [wed] \
[thu] [fri] [sat] [sun]})] \
[time({anytime|*startTime*:*endTime*}) \
|restrictions-}] \
[targuid(*userName*)] \
[trust | trust-] \
[value{+|-}(*value*)] \
[warning | warning-]

ac_id(*id*)

Defines a unique ID for the endpoint (HNODE object) that is saved in the local CA Access Control database and on the DMS. CA Access Control uses this ID to identify the HNODE, so that changes to the endpoint's IP address or name do not affect advanced policy management functionality; CA Access Control can still trace the endpoint.

audit

Indicates which access events are logged. Specify one of the following attributes:

- **all**-CA Access Control logs both authorized and unauthorized access attempts.
- **failure**-CA Access Control logs unauthorized access attempts. This is the default value.
- **none**-CA Access Control does not write any records in the log file.
- **success**-CA Access Control logs authorized access attempts.

calendar(*calendarName*)

Defines Unicenter NSM calendar records that represent time restrictions in Unicenter TNG. CA Access Control maintains a list of these objects for management purposes only, but doesn't protect them. When assigning more than one calendar, separate the calendar names with a space or a comma.

calendar-(*calendarName*)

Deletes one or more Unicenter NSM calendar records from the resource record. Use this parameter with the chres or editres command only.

category(*categoryName* [,*categoryName...*])

Assigns one or more security categories to the resource record.

If you specify the category parameter when the CATEGORY class is not active, CA Access Control updates the resource definition in the database; however, the updated category assignment has no effect until the CATEGORY class is activated again.

category-(*categoryName* [,*categoryName...*])

Deletes one or more security categories from the resource record.

The specified security categories are deleted from the resource record, regardless of whether the CATEGORY class is active. Use this parameter only with the chres or editres command.

className

Specifies the name of the class to which the resource belongs. To list the resource classes defined to CA Access Control, use the find command.

cmd+(*selang_command_string*)

Specifies a list of selang commands that define the policy. These are the commands used to deploy the policy. For example,

```
editres RULESET IIS5#02 cmd+("nr FILE /inetpub/* defaccess(none) owner(nobody)")
```

cmd-

Removes policy deployment command list from the RULESET object.

comment(*string*)

Adds an alphanumeric string of up to 255 characters to the resource record. If the string contains any blanks, enclose the entire string in single quotation marks. The string replaces any existing string defined previously.

Note: For the SUDO class, this string has a special meaning. For more information about defining SUDO records, see the *Endpoint Administration Guide for UNIX*.

comment-

Deletes the comment from the resource record. Use this parameter only with the chres or editres command.

container(*containerName*)

Represents CONTAINER objects, a generic grouping class.

containerName is the name of one or more CONTAINER records defined in the CONTAINER class. When assigning more than one CONTAINER, separate the names with a space or a comma.

container-(*containerName*)

Deletes one or more CONTAINER records from the resource record. Use this parameter with the chres or editres command only.

dates(*time-period*)

Defines one or more periods when users cannot log in, such as holidays. If more than one time period is specified, separate the periods with a space. Use the following format:

mm/dd[/yy[yy]][@hh:mm][-mm/dd]/[/yy[yy]][@hh:mm]

If you do not specify a year, (or you specify a year before 1990), it means the period or holiday is annual. You can specify the year with two digits or four digits, for example: 98 or 1998.

If you do not specify a start time then the start of the day (midnight) is used; if you do not specify an end time then the end of the day (midnight) is used. The format of the hours and the minutes is *hh:mm*, where *hh* is the hour in 24-hour notation (00 through 23) and *mm* is the minutes (00 through 59).

If you do not specify an interval of time (for example, 12/25@14:00-12/25@17:00), but only a day and a month (12/25), then the holiday lasts for one whole day.

If you are issuing the command in a different time zone from where the holiday occurs, translate the period to your local time. For example, if you are in New York and Los Angeles has a half-day holiday, you must enter 09/14/98@18:00-09/14/98@20:00. This prevents the users from logging in from 3:00 p.m. to 5:00 p.m. in Los Angeles.

defaccess([*accessAuthority*])

Defines the default access authority for the resource. The default access authority is the authority granted to any accessor not in the resource's access control list that requests access to the resource. The default access is also applied to users who are not defined in the database. Valid access authority values vary by class.

If you omit *accessAuthority*, CA Access Control assigns the implicit access specified in the UACC property of the record that represents the resource's class in the UACC class.

dh_dr{+|-}(*dh_dr*)

Defines Distribution Hosts this endpoint uses for disaster recovery.

filepath(*filePaths*)

Defines one or more absolute file paths, each of which constitutes a valid kernel module. Multiple file paths are separated by a colon (:).

flags(*flagName*)

Defines how the resource is to be trusted and how to check it for trusted status. Available flags are Ctime, Mtime, Mode, Size, Device, Inode, Crc, and Own/All/None.

gacc(*access-value*)

Lets a program access protected, frequently-opened files at a much faster rate than otherwise possible.

gowner(*groupName*)

Assigns a CA Access Control group as the owner of the resource record. The group owner of the resource record has unrestricted access to the resource, provided the group owner's security level, security label, and security category authorities are sufficient to allow access to the resource. The group owner of the resource is always permitted to update and delete the resource record. See the *Endpoint Administration Guide for UNIX* for more information.

label(*labelName*)

Assigns a security label to the resource record.

label-

Deletes the security label from the resource record. Use this parameter only with the chres or editres command.

level(*number*)

Assigns a security level to the resource record. Enter a positive integer between 1 and 255.

level-

Removes any security level from the resource. Use this parameter only with the chres or editres command.

mask (*IPv4-address*) match (*IPv4-address*)

The *mask* and *match* parameters are applicable only to HOSTNET records. They are required when creating a HOSTNET record and are optional when modifying a record.

Use mask and match together to define the group of hosts defined by a HOSTNET record. A host is a member of a HOSTNET record group if an AND of the host IP address with the mask address produces the match address.

For example, specifying mask(255.255.255.0) and match(192.16.133.0) means a host is a member of the group if it has an IP address in the range 192.16.133.0 to 192.16.133.255.

The mask and match parameters require IPv4 addresses.

mem(*resourceName*)

Adds a member resource to a resource group. If you are adding more than one member resource, separate each name with a comma.

You can use the mem parameter only with resource records of the following classes:

- CONTAINER. This class defines a group of objects from other resource classes.
- GFILE. This class contains resource records that define groups of files.
- GHOST. This class contains resource records that define groups of hosts.
- GSUDO. This class contains resource records that define groups of commands.
- GTERMINAL. This class contains resource records that define groups of terminals.
- GPOLICY. This class contains resource records that define a logical policy.
- GHNODE. This class contains resource records that define a host group.
- GDEPLOYMENT. This class contains resource records that define the policy deployment.

Use the mem parameter to add a record of the appropriate type to a resource group, for example, to add a FILE record to a resource group of class GFILE.

Note: If you are using the mem parameter for CONTAINER resources, you must also include the of_class parameter.

Both the member resource and the resource group must already be defined in CA Access Control. To create a resource group, create a resource of the class you want. For example, the following command creates a GFILE resource group:

```
newres GFILE myfiles
```

mem-(*resourceName*)

Removes member resources from a resource group. If you are removing more than one member resource, separate the resource names with a space or a comma. Use this parameter only with the chres or editres command.

node_alias{-|+}(*alias*)

Defines an endpoint alias.

Defining aliases for the endpoint aliases lets CA Access Control send advanced policy management commands to the actual endpoint based on the alias.

node_ip[-|+](*ip*)

Defines the IP address of the host. Advanced policy management uses the IP address, in conjunction with the endpoint's name, to locate the required endpoint.

notify(*mailAddress*)

Instructs CA Access Control to send notification messages whenever the resource represented by the resource record is accessed. Enter a user name, an email address of a user, or the email address of a mail group if an alias is specified.

Notification takes place only when the Log Routing System is active. The notification messages are sent either to the screen or to the mailbox of the users, depending on the setup of the Log Routing System.

Each time a notification message is sent, an audit record is written in the audit log. For information on filtering and viewing audit records, see the *Endpoint Administration Guide for UNIX*.

The recipient of notify messages should log in frequently to respond to the unauthorized access attempts described in each message.

Limit: 30 characters.

notify-

Specifies that no one is notified when the resource represented by the resource record is successfully accessed. Use this parameter only with the chres or editres command.

of_class(*className*)

Specifies the resource type for the record you are adding to the CONTAINER class with the mem parameter.

owner(*Name*)

Assigns a CA Access Control user or group as the owner of the resource record. The owner of the resource record has unrestricted access to the resource, provided the owner's security level, security label, and security category authorities are sufficient to allow access to the resource. The owner of the resource is always permitted to update and delete the resource record. See the *Endpoint Administration Guide for UNIX* for more information.

password

Specifies, for the SUDO class, that the sesudo command requires the original user's password.

password-

Cancels the password parameter, so that the sesudo command no longer requires the original user's password. Use this parameter with the chres or editres command only. If the password parameter was not used previously, then this parameter is unnecessary.

policy(name(*name#xx*) status(*status*) updated_by(*name*)) |
policy(name(*name#xx*) deviation{+|-})

Adds a subscriber of the node in the propagation tree and specifies its status. Alternatively, updates an existing policy version to specify whether a policy deviation exists or not. The updated_by property must be updated when updating policy status. It is a string representing the name of the user that changed the policy status.

Policy status can be one of Transferred, Deployed, Undeployed, Failed, SigFailed, Queued, UndeployFailed, or TransferFailed.

policy-[(name(*name#xx*))]

Removes the named policy version from the node. If no policy is specified, all policies deployed to this node are removed.

resourceName

Defines the name of the resource record to modify or add. When changing or adding more than one resource, enclose the list of resource names in parentheses and separate the resource names with a space or a comma. At least one resource name must be specified.

CA Access Control processes each resource record independently in accordance with the specified parameters. If an error occurs while processing a resource, CA Access Control issues a message and continues processing with the next resource in the list.

Note: If you use a variable in a resource name, use the following syntax to refer to the variable: <!*variable*>, for example, <!AC_ROOT_PATH>\bin. You can only use variables in selang rules in policies.

restrictions([days] [time])

Specifies the days of the week and the hours in the day when users can access the file.

If you omit the days argument and specify the time argument, the time restriction applies to any day-of-week restriction already indicated in the record. If you omit time and specify days, the day restriction applies to any time restriction already indicated in the record. If you specify both days and time, the users may access the system only during the specified time period on the specified days.

- [Days] specifies the days on which users may access the file. The days argument takes the following sub-arguments:
  - **anyday**-Allow users access to the file on any day.
  - **weekdays**-Allow users access to the resource only on weekdays-Monday through Friday.
  - **Mon**, **Tue**, **Wed**, **Thu**, **Fri**, **Sat**, **Sun**-Allow users access to the resource only on the specified days. You can specify the days in any order. If you specify more than one day, separate the days with a space or a comma.
- [Time] specifies the period during which users may access the resource. The time argument takes the following sub-arguments:
  - **anytime**-Allow users access to the resource at any time of the day.
  - **startTime:endTime**-Allow access to the resource only during the specified period. The format of both startTime and endTime is *hhmm*, where *hh* is the hour in 24-hour notation (00 through 23) and *mm* is the minutes (00 through 59). Note that 2400 is not a valid time value. startTime must be less than endTime, and both times must occur on the same day. If the terminal is in a different time zone from the processor, adjust the time values by translating the start and end times for the terminal to the equivalent local times for the processor. For example, if the processor is in New York and the terminal is in Los Angeles, to allow access to the terminal from 8:00 a.m. to 5:00 p.m. in Los Angeles, specify time (1100:2000).

restrictions-([days] [time])

Deletes any restrictions that limit the users' ability to access the file.

ruleset+(*name*)

Specifies a rule set to associate with the policy.

ruleset-(*name*)

Deletes a rule set from the policy. If no ruleset is specified, removes all rulesets from the policy.

signature(*hash_value*)

Specifies a hash value. For a policy, this is based on signatures of RULESET objects associated with the policy. For a ruleset, this is based on the policy deployment command list and policy undeployment (removal) command list.

subscriber(name(*sub_name*) status(*status*))

Adds a subscriber of the node in the propagation tree and specifies its status. Status can be one of **unknown**, **available**, **unavailable**, or **sync**.

subscriber-(name(*sub_name*)) | sub-

Removes a subscriber database from the node. If no subscriber is specified, all subscribers are removed.

targuid(*userName*)

Specifies, for the SUDO class, the name of the user whose authority is borrowed for executing the command. Default is root.

trust

Specifies that the resource is trusted. The trust parameter applies only to resources of the PROGRAM and SECFILE classes. Users can execute the program as long as the program remains trusted. See the *Endpoint Administration Guide for UNIX* for more information. Use this parameter only with the chres or editres command.

trust-

Specifies that the resource is untrusted. The trust- parameter applies only to resources of the PROGRAM and SECFILE classes. Users cannot execute an untrusted program. See the *Endpoint Administration Guide for UNIX* for more information. Use this parameter only with the chres or editres command.

undocmd+(*selang_command_string*)

Specifies a list of selang commands that define policy undeployment. These are the commands used to remove the deployed policy (undeploy). For example:

```
editres RULESET IIS5#02 undocmd+("rr FILE /inetpub/*")
```

undocmd-

Removes policy removal command list from the RULESET object.

value+(*value*)

Adds the specified value to the specified variable (ACVAR object).

value-(*value*)

Removes the specified value from the specified variable (ACVAR object).

warning

Specifies that, even if an accessor's authority is insufficient to access the resource, CA Access Control is to allow access to the resource. However, CA Access Control writes a warning message in the audit log.

Note: In Warning Mode, CA Access Control does not create warning messages for resource groups.

warning-

Specifies that, if an accessor's authority is insufficient to access the resource, CA Access Control is to deny the user access to the resource and does not write a warning message. Use this parameter only with the chres or editres command.

Examples

- The user *admin1* wants to change the owner and default access for the terminal *tty30* and restrict the use of the terminal to weekdays during regular business hours (8:00 a.m. to 6:00 p.m.).

  – The user admin1 has the ADMIN attribute.

  ```
  chres TERMINAL tty30 owner(admin1) defaccess(read)  restrictions \
  (days(weekdays)time(0800:1800))
  ```

- The admin user Sally wants to remove the group and owner property stored in a FILE class record for file account.txt.

  – The user Sally is the owner of Jared's user record.

  ```
  chres FILE /account.txt group() owner()
  ```

  To remove any record property, if the property is defined by a string, type the property with either the "-" sign or empty parenthesis "()".

- The user Bob wants to delete the comment field of the terminal tty190 and be notified whenever access to the terminal is granted.
  - The user Bob is a CA Access Control user and is the owner of the terminal tty190.

  ```
  chres TERMINAL tty190 comment- notify(Bob@athena)
  ```

- The user Admin1 wants to add the OPERATOR category to the list of security categories of the resource USER.root, which is in the SURROGATE class.
  - The user Admin1 has the ADMIN attribute.
  - The OPERATOR category is defined in the database.

  ```
  chres SURROGATE USER.root category(OPERATOR)
  ```

- The user admin1 wants to define /bin/su as a trusted program with a global access of EXECUTE.
  - The user admin1 has the ADMIN attribute.
  - The following defaults apply:
    - restrictions(days(anyday) time(anytime))
    - owner(admin1)
    - audit(failure)

  ```
  newres PROGRAM /bin/su defaccess(x) trust
  ```

- The user admin1 wants to define the substitution of group ID to the group "system" as a protected resource to which no user, including admin1, has access.
  - The user admin1 has the ADMIN attribute. The user nobody is defined to CA Access Control.
  - The following defaults apply:
    - restrictions(days(anyday) time(anytime))
    - audit(failure)

  ```
  newres SURROGATE GROUP.system defaccess(n) owner(nobody)
  ```

- The user SecAdmin wants to define ProjATerms, a group of terminals containing the terminals T1, T8, and T11. The terminal group is to be used only by the group PROJECTA and only on weekdays during regular business hours (8:00 a.m. to 6:00 p.m.).
  - The user SecAdmin has the ADMIN attribute.
  - The terminals T1, T8, and T11 are defined to CA Access Control.
  - The group PROJECTA is defined to CA Access Control.
  - audit(failure)

  ```
  newres GTERMINAL ProjATerms mem(T1,T8,T11) owner(PROJECTA) \
  restrictions(days(weekdays) time(0800:1800)) defaccess(n)
  ```

More information:

rmres Command—Delete a Resource (see page 130)
showres Command—Display Resource Properties (see page 147)
authorize Command—Set Access Authorities on a Resource (see page 44)
chres Command—Modify Windows Resources (see page 183)
find Command—List Database Records (see page 113)
CONTAINER Class (see page 245)
Access Authority by Class (see page 28)

ch[x]usr Command—Change User Properties

Valid in the AC environment

Use the commands chusr, chxusr, editusr, editxusr, newusr, and newxusr to change the properties of users, and to define the user records in the CA Access Control database if necessary.

These commands all have synonyms, as follows:

- chusr—cu
- chxusr—cxu
- editusr—eu
- editxusr—exu
- newusr—nu
- newxusr—nxu

This means, for example, that the command cu is identical to the command chusr.

All these commands are identical in structure, and vary only in their scope. Use these commands as follows:

- Use the chusr, editusr, and newusr commands for internal users. The differences between these commands are as follows:
  - The chusr command *modifies* one or more USER records.
  - The editusr command *creates or modifies* one or more USER records.
  - The newusr command *creates* one or more USER records.

  Note: These commands also exist in the native environment but operate differently there.
- Use the chxusr, editxusr and newxusr commands for enterprise users. The differences between these commands are as follows:
  - The chxusr command *modifies* one or more XUSER records.
  - The editxusr command *creates or modifies* one or more XUSER records.
  - The newxusr command *creates* one or more XUSER records.

The USER and XUSER class records are identical for all properties, except that where properties are defined in the enterprise user stores, the XUSER records do not redefine them.

When you execute these commands, the changes that you make modify the user record immediately, even if the user is currently logged in to the system.

Authorization Required

The level of authority required to execute these commands depends on which parameters you want to specify. The following rules apply:

- If you have the ADMIN attribute, you can specify all parameters except audit.
- If you have the AUDITOR attribute, you can specify the audit parameter. (The auditor[-] parameter requires the ADMIN attribute.)
- When updating an existing record, the owner of the user record can specify all parameters except admin, auditor, server, operator, and pwmanager. An owner can assign a security category to the user record if the security category is in the owner's user record. An owner can assign a security label to the user record if the security label is assigned in the owner's user record. The owner of the user record can assign any security level that is less than or equal to the security level assigned in the owner's user record.
- If the user record is within the scope of a group in which you have the GROUP-ADMIN attribute, you have the same authority as the owner of the record.
- If the user record is within the scope of a group in which you have the GROUP-AUDITOR attribute, you can specify the audit parameter.
- If in the access control list of the USER record in the ADMIN class, you have MODIFY authority (for ch[x]usr), or CREATE authority (for edit[x]usr), you have the same authority as the owner of the user record.

```
{{chusr|cu}|chxusr|cxu}|{editusr|eu}|{editxusr|eu}|{newusr|nu}| {newxusr|nxu}} \
    {userName|(userName [,userName...])} \
    [{admin | admin-}] \
    [audit({none | all | {[success][failure][loginsuccess]|[loginfail]|[trace]}})] \
    [{auditor | auditor-}] \
    [{category(categoryName) | category-(categoryName)}] \
    [{comment(string) | comment-}] \
    [country(string)] \
    [email(emailAddress)] \
    [enable] \
    epwasown(password) \
    [{expire[(date)] | expire-}] \
    [fullname (fullName)]
    [{gowner(groupName)] \
    [{grace(nLogins) | grace-}] \
    [{ign_hol | ign_hol-}] \
    [{inactive(nDays) | inactive-}] \
    [{interval(nDays) | interval-}] \
    [{label(labelName) | label-}] \
    [{level(number) | level-}] \
    [location(string)] \
    [{logical|logical-}] \
    [{maxlogins(nLogins) | maxlogins-}] \
    [{min_life(nDays) | min_life-}] \
    [{notify(mailAddress) | notify-}] \
    [{operator | operator-}] \
    [organization(string)] \
    [org_unit(string) \
    [owner({userName | groupName})] \
    [password(string)] \
    [phone(string)] \
    [{pmdb(pmdbName) | pmdb-}] \
    [{profile(groupName) | profile-}] \
    [pwasown(string)] \
    [{pwmanager | pwmanager-}] \
    [regular] \
    [{restrictions( \
        [days({anyday|weekdays|[mon] [tue] [wed] [thu] [fri] [sat] [sun]})] \
        [time({anytime|startTime:endTime})]
        ) |restrictions-}] \
    [{resume[(date)] | resume-}] \
    [{server | server-}] \
    [{suspend[(date)] | suspend-}] \
```

```
[nt|nt( ) \
    [admin|admin-] \
    [comment('comment')|comment- ] \
    [country('country-name')] \
    [expire|expire(mm/dd/yy[@hh:mm])|expire-] \
    [flags({account-flags}|-account-flags})] \
    [homedir(any-string)] \
    [homedrive(home-drive)] \
    [location(any-string)] \
    [logonserver(server-name)] \
    [name(full-name)] \
    [organization(name)] \
    [org_unit(name)] \
    [password(user's temporary password)] \
    [pgroup(primary-group)] \
    [phone(any-string)] \
    [privileges(privilege-list)] \
    [restrictions(days(day-data) time(hhmm:hhmm|anytime) )] \
    [script(logon-script-path)] \
    [workstation(workstation-list)] )] \
[unix({ [gecos(string)] \
    [homedir(path)] \
    [pgroup(groupName)] \
    [shell|prog(fileName)] \
    [userid(number)]}]
```

admin

Assigns the ADMIN attribute to the user. A user with the ADMIN attribute is allowed to issue all selang commands with all parameters except the audit parameter. You must have the ADMIN attribute to use the admin parameter.

admin-

Removes the ADMIN attribute from the user. (CA Access Control ensures that at least one user has the ADMIN attribute.)

You cannot use this parameter with the new[x]usr command.

audit

Specifies which user activities on resources protected by CA Access Control are logged to the audit log. To specify more than one event type, separate the event type names with a space or a comma. The audit attributes are as follows:

- **all—**CA Access Control logs all user activities. The monitored activities are: failure, loginfail, loginsuccess, success, and trace.
- **failure—**CA Access Control logs failed access attempts.
- **loginfail—**CA Access Control logs failed login attempts.
- **loginsuccess—**CA Access Control logs successful logins.
- **none—**CA Access Control does not log any user activities.
- **success—**CA Access Control logs successful accesses.
- **trace—**CA Access Control logs every message that appears in the trace file because of this user's actions.

auditor

Assigns the AUDITOR attribute to the user. A user with the AUDITOR attribute can audit the use of system resources and is able to control the logging of detected accesses to any CA Access Control-protected resource during CA Access Control authorization checking and accesses to the database. See the *Endpoint Administration Guide* for your OS for more information on the authorities granted to a user with the AUDITOR attribute.

auditor-

Removes the AUDITOR attribute from the user record.

You cannot use this parameter with the new[x]usr command.

auth_type

Specifies the authentication method.

Used only by SSO.

You cannot use this parameter for enterprise users.

category(*categoryName*[*, categoryName...*])

Assigns one or more security categories to the user.

category-(*categoryName*[*, categoryName...*])

Removes one or more security categories from the user record.

You cannot use this parameter with the new[x]usr command.

comment(*commentString*)

Assigns a comment to the user record.

commentString

Specifies the comment. *commentString* is an alphanumeric string of up to 255 characters. If *commentString* contains blanks, enclose it in single quotation marks.

comment-

Deletes the comment from the user record.

You cannot use this parameter with the new[x]usr command.

country(*countryName*)

Specifies the country where the user is located. The country is not used during the authorization process.

countryName

Defines the country. It is an alphanumeric string of up to 19 characters. If the string contains blanks, enclose the entire string in single quotation marks.

email(*emailAddress*)

Defines the user's email address.

emailAddress

Defines the user's email address.

Limits: Up to 128 characters

enable

Enables the login of a user that has for any reason been disabled.

You cannot use this parameter with the new[x]usr command.

epwasown(*password*)

Changes the user password as if the the user changes their own password. This password change is not an administrative change and so does not automatically expire the password.

expire(*dateTime*)

Sets the date when the user account expires. If a date is not specified, the account expires immediately, or if the user is logged in, when the user logs out.

If the user record has a value for this property, that value overrides the value in the GROUP record.

Note: Use the expire- parameter to enable expired user records; you do not use the resume parameter to do this.

dateTime

Defines the date, and optionally the time. It has the following format: *mm*/*dd*/[*yy*]*yy*[@*HH:MM*]

You can use either two digits or four digits to specify the year.

expire-

For the new[x]usr command, defines a user account that does not have an expiration date.

For the ch[x]usr and edit[x]usr commands, removes an expiration date from a user account.

flags(*accountFlags*|*-accountFlags*)

Specifies particular attributes of a user's account. See the appendix "Windows Values" for a list of valid flag values.

To remove flags from the user record, precede *accountFlags* with a minus (-).

fullname(*fullName*)

Specifies the full name of the user.

fullName

Defines the full name. It is an alphanumeric string of up to 47 characters. If *fullName* contains blanks, enclose the entire string in single quotation marks.

gecos(*string*)

Specifies a comment string for the user. Enclose the string in single quotation marks.

gowner(*groupName*)

Assigns a CA Access Control group as the owner of the user record. The group owner of the user record has unrestricted access to it, provided the group owner's security level and security category authorities are sufficient. The group owner of the user record is always permitted to update and delete the user record.

grace(*nLogins*)

Defines the number of grace logins the user is allowed.

After the number of grace logins is reached, the user is cannot access the system and must contact the system administrator to select a new password. If grace is set to zero, the user cannot log in.

If the user record has a value for this parameter, that value overrides the value in the GROUP record.

If this parameter is not specified and the user has a profile group that contains a value for this parameter, the value in the GROUP record is used. If neither the USER nor GROUP record contains a value, the CA Access Control global grace login setting is used.

nLogins

Defines the number of grace logins. Enter an integer between 0 and 255.

grace-

Deletes the user's grace login setting. The CA Access Control global grace login setting is used instead.

You cannot use this parameter with the newusr command.

homedir(*path*)

Specifies the full path of the user's home directory. If *path* ends with a slash, CA Access Control concatenates *userName* to the path.

homedrive(*drive*)

Specifies the drive of the user's home directory.

ign_hol

Assigns the IGN_HOL attribute to the user. A user with the IGN_HOL attribute can log in during any period defined in a holiday record.

ign_hol-

Removes IGN_HOL attribute from the user.

inactive(*nDays*)

Specifies the number of days that must pass before the system changes the user to inactive. When the number of days is reached, the user cannot log in.

Note: Inactive users are not marked in the user record. To identify inactive users, you must compare the Last Accessed Time value with the Inactive Days value.

nDays

Defines the number of days. *nDays* is zero or a positive integer. If *nDays* is zero, the effect is the same as using the inactive- parameter.

inactive-

Changes the user's status from inactive to active.

You cannot use this parameter with the newusr command.

interval(*nDays*)

Defines the number of days that must pass after the password was set or changed before the system prompts the user for a new password. Enter zero or a positive integer. If *nDays* is zero CA Access Control disables password interval checking and the password does not expire. This means the default set by the setoptions command is not used. Set *nDays* to zero only for users with low security requirements.

When *nDays* is reached, CA Access Control informs the user that the password has expired. The user can continue to use the password until the number of grace logins is reached. After the number of grace logins is reached, the user is denied access to the system and must contact the system administrator to be given a new password.

interval-

Cancels a user's password interval setting. If the user has a profile group with a value for this parameter, that value is used. Otherwise, the default set by the setoptions command is used.

You cannot use this parameter with the new[x]usr command.

label(*labelName*)

Assigns a security label to the user.

label-

Deletes the security label from the user record.

You cannot use this parameter with the new[x]usr command.

level(*levelNumber*)

Assigns a security level to the user record.

levelNumber is an integer between 0 and 255.

level-

Deletes the security level from the user record,

You cannot use this parameter with the newusr command.

localapps

Used by CA SSO.

location(*locationString*)

Specifies the user's location. The location is not used during the authorization process.

locationString

Defines the location. *locationString* is an alphanumeric string of up to 47 characters. If *locationString* contains blanks, enclose it in single quotation marks.

logical

Assigns the LOGICAL attribute to the user. A user with the LOGICAL attribute cannot log in and is used for internal CA Access Control purposes only.

For example, the user nobody that you can use as the owner of resources to prevent even the resource owner from accessing the resource is a logical user by default. This means that no user can log in using this account.

logical-

Removes the LOGICAL attribute from the user.

logonserver(*server-name*)

Specifies the server that verifies the login information for the user. When the user logs in to the domain workstation, CA Access Control transfers the login information to the server, which gives the workstation permission for the user to work.

maxlogins(*nLogins*)

Sets the maximum number of concurrent logins for the user. A value of 0 (zero) means that the user can log in from any number of terminals concurrently. If this parameter is not specified, the global maximum logins setting is used.

Note: If maxlogins is set to 1, you cannot run selang. You must shut down CA Access Control, change the maxlogins setting to greater than one, for example by using setpropadm utility, and start CA Access Control again.

maxlogins-

Deletes the user's maximum login setting. The global setting is used instead.

You cannot use this parameter with the new[x]usr command.

min_life(*nDays*)

The minimum number of days that must pass before the user is allowed to change the password again. Enter a positive integer.

min_life-

Deletes the user's min_life setting. If the user has a profile group with a value for this parameter, that value is used. Otherwise, the default set by the setoptions command is used.

You cannot use this parameter with the new[x]usr command.

nochngpass

Specifies that the user is not allowed to change passwords for another user.

notify(*notifyAddress*)

Sends an email to *notifyAddress* every time the user logs in. The recipient of the notify messages should log in frequently to respond to the unauthorized access attempts described in each message.

When CA Access Control sends a notification message, it writes an audit record in the audit log.

notifyAddress

Defines a user name or an email address.

Limit: 30 characters.

notify-

Specifies that no one is notified when the user logs in.

You cannot use this parameter with the new[x]usr command.

nt

For the chusr and editusr commands, this parameter changes the user's definition in the local Windows system.

For the newusr command, this parameter adds the user to the local Windows system.

If more than one argument is specified, separate the arguments with a space.

See the environment command, for more information on how to operate on the local Windows system from within CA Access Control.

The nt option, and sub-options under the nt option, are not valid for enterprise users.

operator

Assigns the OPERATOR attribute to the user. A user with the OPERATOR attribute can list all resource records in the database, and has read authority for all CA Access Control defined files.

A user with this attribute can also use all the options of the secons command. See the *Reference Guide* for more information on the secons utility.

operator-

Removes the OPERATOR attribute from a user record.

You cannot use this parameter with the newusr command.

organization(*organizationString*)

Specifies the user's organization. The organization is not used during the authorization process.

organizationString

Defines the organization. *organizationString* is an alphanumeric string of up to 255 characters. If *organizationString* contains blanks, enclose it in single quotation marks.

org_unit(*org_unitString*)

Specifies the user's organization unit. The organization unit is not used during the authorization process.

org_unitString

Defines the organization unit. *org_unitString* is an alphanumeric string of up to 255 characters. If *organizationString* contains blanks, enclose it in single quotation marks.

owner(*Name*)

Assigns a CA Access Control user or group as the owner of the user record. See the *Endpoint Administration Guide* for your OS for more information.

password(*string*)

Assigns a password to a user. Specify any character except a space or a comma. If password checking is enabled, the password is valid for one login only. When the user next logs in to the system, a new password must be set.

To change your own password, you need to set selang options using *setoptions cng_ownpwd* or use sepass.

pgroup(*groupName*)

Sets the user's primary group ID. *groupName* is the name of a UNIX group.

phone(*phoneString*)

Defines the user's phone number. The phone number is not used during the authorization process.

phoneString

Defines the phone number. *phoneString* is an alphanumeric string of up to 19 characters. If *phoneString* contains blanks, enclose it in single quotation marks.

pmdb(*pmdbName*)

Specifies that when a user changes a password with the sepass utility, the new password is propagated to the specified PMDB. Enter the fully qualified name of the PMDB. The password is not sent to the Policy Model defined in the parent_pmd or passwd_pmd tokens in the [seos] section of seos.ini.

This option cannot be used for enterprise users.

pmdb-

Removes the PMDB attribute from the user record.

You cannot use this parameter with the new[x]usr command.

privileges(*privilege-list*)

Adds specific rights to the Windows user record or, when privList is preceded by a minus sign (-), removes the specified rights.

You cannot use this parameter with the newusr command.

profile(*groupName*)

Assigns a user to a profile group. The following values can be taken from the profile group:

- audit
- auth_type
- expire
- grace
- inactive
- interval
- maxlogins
- min_life
- password rules
- pmdb
- pwd_autogen
- pwd_policy
- pwd_sync
- restrictions (days, time)
- resume
- suspend
- unix (homedir, shellprog)

profile-

Removes a user from the profile group.

You cannot use this parameter with the new[x]usr command.

pwmanager

Assigns the PWMANAGER attribute to the user. A user with this attribute can change the passwords of users in the database. See the *Endpoint Administration Guide* for your OS for more information.

pwmanager-

Removes the PWMANAGER attribute from the user record.

You cannot use this parameter with the new[x]usr command.

pwasown(*string*)

Replaces a password as if changed by the user. Specifying this parameter updates the time and date of the last change in the database. Grace logins are terminated.

regular

Resets the OBJ_TYPE property of the record, and so removes authority attributes from the user.

restrictions([*Days*] [*Time*])

Specifies the days of the week and the times in the day when users can be logged in. The restrictions are stored in the DAYTIME property of the [X]USER record.

If you omit *Days* and specify *Time*, the time restriction applies to any day-of-week restriction that is already defined in the record.

If you omit *Time* and specify *Days*, the *Days* restriction applies to any time restriction already defined in the record.

If you specify both *Days* and *Time*, the users can access the system only during the specified time period on the specified days.

Days

Specifies the days on which users can be logged in. You can use the following keywords when you specify *Days:*

- **anyday—**Allow users access to the file on any day.
- **weekdays—**Allow users access to the resource only on weekdays-Monday through Friday.
- **Mon**, **Tue**, **Wed**, **Thu**, **Fri**, **Sat**, **Sun—**Allow users access to the resource only on the specified days. You can specify the days in any order. If you specify more than one day, separate the days with a space or a comma.

Time

Specifies the period during which users can be logged in. The time argument takes the following sub-arguments:

- **anytime**—Allow users access to the resource at any time of the day.
- *startTime*:*endTime*—Allow access to the resource only during the specified period.

  The format of *startTime* and *endTime* is *hhmm*, where *hh* is the hour (00 through 23) and *mm* is the minutes (00 through 59). Note that 2400 is not a valid time value; use 0000 instead.

  startTime must be less than *endTime*.

  Note: CA Access Control uses the time zone of the processor. If the user logs in at a terminal in a different time zone from the processor, you must take this into account.

restrictions-([days] [time])

Deletes any restrictions that limit the users' ability to be logged in.

resume([*dateTime*])

Enables a user record that was disabled by specifying the suspend parameter. If you specify both the suspend parameter and the resume parameter, the resume date must fall after the suspend date. If you omit *dateTime*, the user record is resumed immediately upon execution of the chusr command. See the *Endpoint Administration Guide* for your OS for more information.

Enter *dateTime* in the format *[m]m/[d]d/yy[@HH:MM].*

resume-

Erases the resume date, and time if used, from the user record. Consequently, the status of the user is changed from active (enabled) to suspended.

You cannot use this parameter with the new[x]usr command.

script(*logon-script-path*)

Specifies the location of a file that runs automatically when the user logs in. This parameter is optional. Typically, this login script configures the working environment. You can also use the profile parameter to set up the user's working environment.

server

Sets the SERVER attribute on. This attribute allows a process running on behalf of the current user to ask for authorization for other users. See the *Endpoint Administration Guide* for your OS for more information.

This option cannot be used for enterprise users.

server-

Sets the SERVER attribute off.

You cannot use this parameter with the new[x]usr command.

shellprog(*fileName*)

Specifies the full path of the initial program or shell that is executed after the user invokes the login or su command. *fileName* is a character string.

This option cannot be used for enterprise users.

suspend([*dateTime*])

Disables a user record, but leaves it defined in the database. A user cannot use a disabled user account to log in to the system.

If *dateTime* is specified, the user record is disabled on the specified date. If *dateTime* is omitted, the user record is disabled immediately upon execution of the ch[x]usr command.

Enter *dateTime* in the format *mm/dd/yy[@HH:MM].*

suspend-

Erases the suspend date from the user record, changing the status of the user from disabled to enabled (active).

You cannot use this parameter with the new[x]usr command.

unix

For the chusr and editusr commands, this parameter changes the user's definition in the local UNIX system.

For the newusr command, this parameter adds the user to the local UNIX system.

If more than one argument is specified, separate the arguments with a space.

See the environment command in this chapter for more information on how to operate on the local UNIX system from within CA Access Control.

The unix option, and sub-options under the unix option are not valid for enterprise users.

userid(*number*)

Sets the user's unique numeric ID (UID), used for unique discretionary access control. *number* is a decimal number. By default, numbers less than 100 are not accepted. See the AllowedGidRange token in the appendix *Reference Guide* for more information about excluded numbers.

***userName*|(*userName* [*,userName...*])**

Defines the name or names of the user or users. Each user name must be unique.

When using the newusr command, *userName* identifies a new user to CA Access Control. If you are using the newusr command and the user is already defined to the native environment, this username will be used by CA Access Control as the USER record that corresponds to that user. Typically, however, you should take advantage of the CA Access Control ability to use enterprise users, and not use newusr to create a USER record for a username that already exists in the native environment. Instead, use the chgxusr command to change the CA Access Control properties of that user.

Sometimes you may want a CA Access Control user name that is not a native login name. In that case, the login command could not put that user to work, but another command such as sesu could.

Note: ON UNIX, where a user name includes a blackslash, use two backslashes when specifying *userName*.

Examples

- The user Bob wants to add the FINANCIAL category to Jim's record, change Jim's security level to 155, and restrict Jim's access to the system to weekdays between 8:00 a.m. and 8:00 p.m.
  - The user Bob has the ADMIN attribute.
  - The user Jim is defined to CA Access Control.
  - The FINANCIAL category is defined to CA Access Control.

  ```
  chuxsr Jim category(FINANCIAL) level(155) restrictions \
  (days(weekdays)time(0800:2000))
  ```

- The user "admin" wants to suspend the user Joel, who will be on vacation for three weeks, starting on August 5, 1995.
  - The user admin has the ADMIN attribute.
  - The user Joel is defined to CA Access Control.
  - Today's date is August 3, 1994.

  ```
  chxusr Joel suspend(8/5/95) resume(8/26/95)
  ```

- The user Security2 wants to remove the AUDITOR attribute from the user Bill and wants to audit all activity by Bill.
  - The user Security2 has the ADMIN and AUDITOR attributes.
  - The user Bill is defined to CA Access Control.

  chxusr Bill auditor audit(all)
- The user Rob wants to change the comment stored in the record of the user Mary.
  - The user Rob is the owner of Mary's user record.

  chxusr Mary comment ('Administrator of the SALES group')
- The admin user Sally wants to remove the country name and the location properties stored in the record of the user Jared.
  - The user Sally is the owner of Jared's user record.

  chxusr Jared country() location()
- The user Bob wants to define the users Peter and Joe to CA Access Control.
  - The user Bob has the ADMIN attribute.
  - The users Peter and Joe are not defined to CA Access Control.
  - The following defaults apply:
    - owner(Bob)
    - audit(failure,loginfailure)

  newusr (Peter Joe)
- The user Bob wants to define the user Jane to CA Access Control and assign "payroll" as the owning group.
  - The user Bob has the ADMIN attribute.
  - The user Jane is not defined to CA Access Control.
  - The full name of the user Jane is JG Harris.
  - audit(failure,loginfailure)

  newusr Jane owner(payroll) name('J.G. Harris')

- The user Bob wants to define the user *JohnD* to CA Access Control with the security category NewEmployee and a security level of three. JohnD is to be allowed to use the system only on weekdays between the hours of 8:00 a.m. and 6:00 p.m.
  - The user Bob has the ADMIN attribute.
  - The NewEmployee category is defined to CA Access Control.
  - The new user's full name is John Doe.
  - The following defaults apply:
    - owner(Bob)
    - audit(failure)

```
newusr JohnD name('John Doe') category(NewEmployee) level(3) \
restrictions(days(weekdays) time(0800:1800))
```

deploy Command—Initiate Policy Deployment

Valid in the AC environment

Use the deploy command to initiate policy deployment. The command executes selang commands stored with the RULESET object that is associated with the POLICY object you are deploying. These are policy deployment commands.

Important! We strongly recommend that you use the policydeploy utility to deploy a stored policy. The deploy command only executes part of the policy deployment and does not update the DMS when deploying a policy to an endpoint.

To run the deploy command, you need to have:

- Sub-administration rights for the POLICY, HNODE, and RULESET classes on each database in the hierarchy below the database where you deploy the policy.
- Appropriate sub-administration rights on each database in the hierarchy below the database where you deploy the policy.

  These are the permissions necessary to execute the selang commands that form the policy on each of these computers.

  For example, you'll need sub-administration rights for the FILE class if you are creating a new file resource:

  ```
  nr FILE /inetpub/* defaccess(none)
  ```

Note: For more information about policy deployment, see the *Enterprise Administration Guide*.

This command has the following format:

deploy POLICY *name#xx*

name#xx

The name of the POLICY object (policy name and version number) for the policy you want to deploy.

deploy- Command—Initiate Policy Removal

Valid in the AC environment

Use te deploy- (or undeploy) command to initiate policy undeployment. The command executes selang commands stored with the RULESET object that is associated with the POLICY object you are deploying. These are policy undeployment commands.

Important! We strongly recommend that you use the policydeploy utility to undeploy a policy. The deploy- command only executes part of the policy undeployment and does not update the DMS when undeploying a policy from an endpoint.

To run this command, you need to have:

- Sub-administration rights for the POLICY, HNODE, and RULESET classes on each database in the hierarchy below the database where you undeploy the policy.
- Appropriate sub-administration rights on each database in the hierarchy below the database where you indeploy the policy.

  These are the permissions necessary to execute the selang commands that form the policy undeployment script on each of these computers.

Note: For more information about deploying a policy, see the *Enterprise Administration Guide*.

This command has the following format:

{deploy-|undeploy} POLICY *name#xx*

name#xx

The name of the POLICY object (policy name and version number) for the policy you want to undeploy.

editfile Command—Create and Modify File Records

Valid in the AC environment

This command is documented with the chfile command.

More information:

chfile Command—Modify File Records (see page 56)

edit[x]grp Command—Create and Modify Group Records

Valid in the AC environment

This command is documented with the ch[x]grp command.

More information:

ch[x]grp Command—Change Group Properties (see page 62)

editres Command—Modify Resource Records

Valid in the AC environment

This command is documented with the chres command.

More information:

chres Command—Modify Resource Records (see page 74)

edit[x]usr Command—Modify User Records

Valid in the AC environment

This command is documented with the chxusr command.

More information:

ch[x]usr Command—Change User Properties (see page 90)

end_transaction Command—Complete Recording Dual Control Transactions

Valid on UNIX hosts in the AC environment

The end_transaction command completes the start_transaction command for Dual Control PMDB processes.

environment Command—Set the Security Environment

Valid in all environments

The environment command sets the security environment. CA Access Control supports the CA Access Control and UNIX security environments. When the selang command shell is invoked, the AC environment is selected by default.

This command has the following format:

environment {ac|config|etrust|native|nt|pmd|seos|unix}

ac

Specifies the CA Access Control security environment. The selang commands affect the local CA Access Control database. Some commands support simultaneous updates to the native OS security settings of the host you are connected to. In the CA Access Control environment, the selang prompt is as follows:

AC>

config

Specifies the remote configuration environment, which lets you change endpoint configuration settings.

etrust

Specifies the CA Access Control security environment.

Note: This is the same as specifying *AC* and is maintained for compatibility with older versions.

native

Specifies the native operating system security environment (either Windows or UNIX) of the host you are connected to, whether local or remote. The selang commands affect the native OS database. In the native environment, the selang prompt is:

AC(native)>

nt

Specifies the Windows security environment. The selang commands affect the Windows database. Some commands support simultaneous updates to the CA Access Control security settings. In the Windows environment, the selang prompt is:

AC(nt)>

pmd

Specifies the selang commands in the remote management environment. When the selang command shell is set to the pmd environment, the commands operate on the PMDB of the selected host. In the pmd environment, the selang prompt is as follows:

AC(pmd)>

seos

Specifies the CA Access Control security environment.

Note: This is the same as specifying *AC* and is maintained for compatibility with older versions.

unix

Specifies the UNIX security environment. The selang commands operate on the UNIX security system. In the UNIX environment, the selang prompt is as follows:

AC(unix)>

find Command—List Database Records

Valid in AC and native environments

The find command displays the names of records in a specified class. If you do not specify any parameters, it displays the names of all classes.

Note: The find command is identical to the commands *list* and *search*.

To use this command you must have sufficient authority, as defined by the following conditions:

- If you have the ADMIN, AUDITOR, or OPERATOR attribute, you can use the find command with all parameters.
- If you have READ authority for a record in the ADMIN class, you can specify the class parameter for the class represented by the record.

This command has the following format:

{find|f|list|search} [{*className*|class(*className*)}] [*objName*]]

className

Specifies the class within which *find* searches for records. If *className* is not provided, *find* lists all classes.

objName

Specifies the records that CA Access Control searches for. *objName* can include wildcard characters.

Example: Display all Records in the TERMINAL Class

To display all the members in the TERMINAL class, enter the following command:

```
find terminal
```

get dbexport Command—Retrieve Exported Database Rules

Valid in the AC environment

The get dbexport command retrieves the rules that were exported from the CA Access Control database or PMD database on the host you are connected to. For the exported database to exist, you must issue the start dbexport command before you issue the get dbexport command.

This command has the following format:

```
get dbexport [pmdname(name)] [params(OFFSET=number)]
```

pmdname(*name*)

(Optional) Defines the name of the PMD database that you exported.

params(OFFSET=*number*)

(Optional) Defines the offset for retrieving more lines from the database output. The get dbexport command can only return 200 lines from the exported database per request. If there is more information in the output, the command returns offset data that specifies the last line returned.

Example: Retrieve rules from an exported database

The following example shows how the get dbexport command is used to retrieve information from the exported CA Access Control database on the host you are connected to. The first command retrieves the first 200 lines and the second command retrieves the following 200 lines of the output:

```
AC > get dbexport
(localhost)
Data for DBEXPORT 'seosdb'
----------------------------------------
setoptions class+(CLASS)
setoptions class+(CLASS)
setoptions class+(CLASS)
...
chres CLASS ("resource") defaccess(none)
OFFSET:  201

AC> get dbexport params("offset=201")
(localhost)
Data for DBEXPORT 'seosdb'

----------------------------------------
chres CLASS ("resource") defaccess(none)
chres CLASS ("resource") defaccess(none)
chres CLASS ("resource") defaccess(none)
...
chres CLASS ("resource") defaccess(none)
OFFSET:  401
```

More information:

start dbexport Command—Initiate Database Export (see page 152)

get devcalc Command—Retrieve Policy Deviation Data

Valid in the AC environment

The get devcalc command retrieves information from the policy deviation data file (deviation.dat) that contains policy deviation calculation results and sends it to one or more set DMS databases. For the data file to exist, the start devcalc command must have been issued before.

When you create a policy or host report, you can also specify to include deviation calculation results. The reporting utility then issues this command.

Important! The deviation calculation does not check whether native rules are applied. It also ignores rules that remove objects (user or object attributes, user or resource authorization, or actual users or resources) from the database. For example, the calculation cannot verify whether the following rule is applied:
rr SUDO admCommand

Note: For more information about the policy deviation data file and advanced policy reporting, see the *Enterprise Administration Guide*.

To run the get devcalc command you must have terminal access rights to the computer and read access to DEVCALC sub-administration class.

This command has the following format:

get devcalc [params("offset=*number*")]

offset=*number*

(Optional) Defines the offset for retrieving more lines from the policy deviation data file. The get devcalc command can only return a maximum number of lines (set by the max_lines_request configuration setting) from the policy deviation data file per request. If there is more information in the file, the command returns offset data that specifies the last line returned.

Example: Get policy deviation data

The following example shows how the get devcalc command is used to retrieve information from the policy deviation data file when the max_lines_request setting is set to 10. The first command retrieves the first ten lines and the second command then retrieves the following ten lines of the output:

```
AC> get devcalc
(localhost)
Data for DEVCALC 'deviation'
-----------------------------------------
DATA       : DATE, Mon Mar 20 11:22:15 2006
POLICYSTART, myPolicy#01
DIFF, (FILE), (file1), (*), (*)
DIFF, (FILE), (file2), (*), (*)
DIFF, (FILE), (file3), (*), (*)
DIFF, (FILE), (file4), (*), (*)
DIFF, (FILE), (file5), (*), (*)
DIFF, (FILE), (file6), (*), (*)
DIFF, (FILE), (file7), (*), (*)
OFFSET   : 11

AC> get devcalc params("offset=11")
(localhost)
Data for DEVCALC 'deviation'
-----------------------------------------
DATA       : DIFF, (FILE), (file8), (*), (*)
DIFF, (FILE), (file9), (*), (*)
DIFF, (FILE), (file10), (*), (*)
DIFF, (FILE), (file11), (*), (*)
DIFF, (FILE), (file12), (*), (*)
DIFF, (FILE), (file13), (*), (*)
DIFF, (FILE), (file14), (*), (*)
DIFF, (FILE), (file15), (*), (*)
DIFF, (FILE), (file16), (*), (*)
DIFF, (FILE), (file17), (*), (*)
OFFSET   : 21
```

More information:

start devcalc Command—Initiate Policy Deviation Calculation (see page 153)
setoptions Command—Set CA Access Control Options (see page 135)

help Command—Get selang Help

Valid in all environments

The help command displays selang syntax in several ways:

- Used without parameters, it displays a list of the selang commands, with a brief explanation of each.
- Used with a selang command name, it displays the syntax of the given command.
- Used with the access parameter, it displays a list of values for the access parameter of the authorize command and the defaccess parameter of the new*, ch*, and edit* commands.
- Used with the lineedit parameter, it displays a list of special characters for selang command line manipulations.

Note: To display the help text for a command typed in the command line without deleting the text in the command line, type Ctrl+2.

{help|h} [*commandName*|access|lineedit|*className*|properties|privilege]

access

Requests a class-by-class list of the access types that the access and defaccess parameters can specify.

className

Requests a short description of the specified class.

command-name

Requests the syntax for the specified command.

lineedit

Requests a list of special characters for selang command line manipulations.

properties

(AC environment) Requests information on how to update user-defined properties.

privilege

(Windows environment) Requests a list of possible Windows privileges for the ch[x]grp, ch[x]usr, edit[x]grp, and edit[x]usr commands.

More information:

selang Commands Reference (see page 37)
selang Environments (see page 33)
Get selang Help (see page 36)

history Command—Show Previously Issued Commands

Valid in all environments

The history command lists all the commands that were entered during the current selang command shell session. The commands are ordered chronologically. Each command is preceded by the number of the command. For example, the third command entered is preceded by the number three.

The history command does not display a password even if one was entered as part of a ch[x]usr, new[x]usr, or edit[x]usr commands. The history command displays a series of asterisks (***) instead of the clear text password.

This command has the following format:

```
history
```

More information:

Command History (see page 23)

hosts Command—Connect to a Remote CA Access Control Terminal

Valid in all environments

The hosts command specifies the hosts or Policy Models to which the selang commands are sent. It lets you connect to a remote CA Access Control computer with a different name, so you can remotely manage the computer while local CA Access Control services are running. By default, all selang commands are directed to the database on the local host.

The hosts command must be executed before executing the commands that are to be directed to the hosts.

To administer (update) a remote host database from the local host, you must meet *one* of the following criteria:

- Be explicitly authorized to update the remote host database from the local database
- Be a member of a group that is allowed to update the remote host database from the local database
- Be the owner of the local host as defined in the remote host

To list all the hosts and PMDBs that are currently available, specify the hosts command without any parameters.

Note: CA Access Control protects hosts through their canonical host names and not through aliases. To avoid the confusion caused by alias names, CA Access Control issues a warning when a HOST rule is defined for an alias name. Similarly, CA Access Control gives a warning if a HOST is defined with less than a fully qualified name, because CA Access Control uses fully qualified names (for example, mymachine.yourcompany.com) for hosts.

This command has the following format:

hosts [{*systemIds*|*policyModel*@[*hostname*]}]

systemIds

Specifies the system IDs of the hosts on which the selang commands are to be executed. When specifying more than one host, enclose the list of systems IDs in parentheses and separate the system IDs with a space or a comma.

***policyModel*@[*hostname*]**

Specifies the addresses of the Policy Models on which the selang commands are to be executed. When specifying more than one Policy Model, enclose the list of Policy Model addresses in parentheses and separate the Policy Model addresses with a space or a comma.

If you do not specify *hostname*, CA Access Control tries to connect to the PMDB on the local host.

Note: The advantage of using a Policy Model over explicitly specifying the hosts is that the system where the Policy Model resides keeps on trying to update all the systems defined to the Policy Model, even if they are currently unavailable. For more information about Policy Models, see the *Endpoint Administration Guide* for your OS.

Example: Let a User or Group Update Remote Hosts

To give a user authorization to update the remote host database from the local database, on the remote host enter the command:

```
authorize TERMINAL local_host uid user_name access(write)
```

To give a group authorization to update the remote host database from the local database, on the remote host enter the command:

```
authorize TERMINAL local_host gid(group_name) access(write)
```

Example: Apply selang Commands to a Remote Policy Model

To apply all subsequent commands to the Policy Model on the station h1, type the command:

```
hosts Policy@h1
```

If the connection to *Policy@h1* is successful, the following message appears.

```
Successfully connected to h1
```

All commands entered from now on are directed to Policy@h1 and not to the local host. The selang prompt changes to the following:

```
Remote_AC>
```

Example: Apply selang Commands to a Remote Host

To apply all future commands to the station athena, type the command:

```
hosts athena
```

If successful connections are made to athena, the following messages appear on the screen.

```
(athena)
Successfully connected
INFO: Target version is 2.50
```

Any command you enter is applied to athena and is not sent to the local host. If you add a new user, the user is only added to athena, as shown in this example:

```
Remote_AC>newusr steve
(athena) USER steve successfully added.
```

join[x] Command—Add Users to Internal Groups

Valid in the AC environment

The join[x] command adds users to one or more internal groups, or changes the users' properties with respect to the groups. The specified users and groups must already be defined to CA Access Control.

Use join to add internal users to groups.

Use joinx to add enterprise users to groups.

Note: This command also exists in the native environment but operates differently there.

The set of properties from the join command *completely replaces* any previous set of properties for the specified users in the specified groups. If any such properties were defined earlier, they are not retained unless the new join command specifies them again.

Note: For more information about group properties, see the *Endpoint Administration Guide* for your OS.

You can use the join command if at least one of the following conditions is true:

- You have the ADMIN attribute.

  Note: If you want to modify CA Access Control GROUP records *and* enterprise groups you need both the MODIFY and JOIN access authority.
- The group record is within the scope of a group in which you have the GROUP-ADMIN attribute.
- You are the owner of the group.
- You are assigned CONNECT authority in the access control list of the GROUP record in the ADMIN class.

This command has the following format:

{join[x]|j[x]} {*userName*|(*userName* [,*userName*...])} \
group(*groupName* [,*groupName*...]) \
[admin|admin-] \
[auditor|auditor-] \
[gowner(*group-name*)] \
[operator|operator-] \
[owner(*userName*|*groupName*)] \
[pwmanager | pwmanager-] \
[regular] \
[nt | unix]

admin

Assigns the GROUP-ADMIN attribute to the user specified by *userName*.

admin-

Removes the GROUP-ADMIN attribute from the user.

auditor

Assigns the GROUP-AUDIT attribute to the user specified by *userName*.

auditor-

Removes the GROUP-AUDIT attribute from the user.

gowner(*groupName*)

Specifies that the user is being added to the group *groupName*.

group(*groupName* [,*groupName...*])

Specifies the group or groups to which the that the user is being added as a member.

nt

Connects *userName* to a group in the Windows database.

operator

Assigns the GROUP-OPERATOR attribute to the user specified by *userName*.

operator-

Removes the GROUP-OPERATOR attribute from the user.

owner(*Name*)

Specifies a CA Access Control user or group as the owner of the join record. If you are creating a connection and you do not specify an owner, you are the owner of the connection.

pwmanager

Assigns the GROUP-PWMANAGER attribute to the user specified by *userName*.

regular

Resets the administrative flags for the user.

unix

Connects *userName* to the group in the UNIX security system.

userName

Specifies a user who is to be connected (or reconnected with a new set of properties) to the group or groups specified by the group parameter.

If the command is join, *userName* is the name of a USER record. If the command is joinx, *userName* is the name of an enterprise user.

Examples

- The user Rorri wants to join the user Bob to the internal group staff.
  - Rorri has the ADMIN attribute.
  - The following defaults apply:
    - admin
    - auditor
    - owner(Rorri)
    - pwmanager

```
join Bob group(staff)
```

- The user Rorri wants to change the definition of Sue in the group staff. She currently is a GROUP-AUDITOR; Rorri wants to add the GROUP-PWMANAGER attribute.
  - Rorri has the ADMIN attribute.
  - The following defaults apply:
    - admin
    - owner(Rorri)

```
join Sue group(staff) auditor pwmanager
```

When selang executes this command, it deletes the previous record. No record is kept of Sue's previous attributes. Therefore, Rorri must specify the two attributes Sue should have now.

More information:

join[x]- Command—Remove Users from Groups (see page 125)
show[x]grp Command—Display Group Properties (see page 145)
show[x]usr Command—Display User Properties (see page 149)

join[x]- Command—Remove Users from Groups

Valid in the AC environment

The join[x]- command removes users from internal groups.

join- removes internal users from internal groups.

joinx- removes enterprise users from internal groups.

Note: The join[-] command also exists in the native environment but operates differently there.

To use the join[x]- command, one of the following conditions must be true:

- You have the ADMIN attribute.

  Note: If you want to modify CA Access Control GROUP records *and* native groups you need both the MODIFY and JOIN access authority.
- The group record is within the scope of a group in which you have the GROUP-ADMIN attribute.
- You are the owner of the group.
- You are assigned CONNECT authority in the access control list of the GROUP record in the ADMIN class.

This command has the following format:

{join[x]-|[x]-} {*userName*|(*userName* [,*userName*...])} \
group(*groupName* [,*groupName*...])

group(*groupName* [,*groupName*...])

Specifies the group or groups from which to remove the user.

userName

Specifies the user you want to remove from the group.

If the command is join, *userName* is the name of a USER record.

If the command is joinx, *userName* is the name of an enterprise user.

Example

The user Bill wants to remove the users sales25 and sales43 from the group PAYROLL.

The user Bill has the ADMIN attribute.

```
joinx- (sales25 sales43) group(PAYROLL)
```

More information:

join[x] Command—Add Users to Internal Groups (see page 122)
show[x]grp Command—Display Group Properties (see page 145)
show[x]usr Command—Display User Properties (see page 149)

list Command—List Database Records

Valid in AC and native environments

This is identical to the find command.

More information:

find Command—List Database Records (see page 113)

newfile Command—Create File Records

Valid in the AC environment

This command is documented with the chfile command.

More information:

chfile Command—Modify File Records (see page 56)

new[x]grp Command—Create Group Records

Valid in the AC environment

This command is documented with the chgrp command.

More information:

ch[x]grp Command—Change Group Properties (see page 62)

newres Command—Create Resource Records

Valid in the AC environment

This command is documented with the chres command.

More information:

chres Command—Modify Resource Records (see page 74)

new[x]usr Command—Create User Records

Valid in the AC environment

This command is documented with the ch[x]usr command.

More information:

ch[x]usr Command—Change User Properties (see page 90)

rename Command—Rename a Database Record

Valid in the AC environment

Renames a record name in the database. The record is now known by its new name only.

Note: You cannot rename records in the SEOS, UACC, and ADMIN classes.

To use the rename command, you must have sufficient authority over the record. CA Access Control makes the following checks until one of the conditions is met:

- You have the ADMIN attribute.
- The resource record is within the scope of a group in which you have the GROUP-ADMIN attribute.
- You are the owner of the record.
- You are assigned CREATE (for editres) access authority in the access control list of the resource class's record in the ADMIN class.

This command has the following format:

rename *className oldresourceName newresourceName*

className

Defines the class to which the record you want to rename belongs.

oldresourceName

Defines the current name of the record in CA Access Control.

newresourceName

Defines the new name you want to assign to the record.

Example

The user ADMIN 1 wants to rename the record *spree3* in class Host to *spree4*.

- The security administrator has the ADMIN attribute.

```
rename host spree3 spree4
```

rmfile Command—Delete File Records

Valid in the AC environment

The rmfile command deletes records belonging to the FILE class from the database.

You can delete a file record if one of the following conditions is met:

- You have the ADMIN attribute.
- The record is within the scope of a group in which you have the GROUP-ADMIN attribute.
- You are the owner of the file.
- You have the DELETE access authority assigned in the ACL of the FILE record in the ADMIN class.

This command has the following format:

```
{rmfile|rf} {fileName | (filename [,filename...])}
```

fileName

Defines the file you are removing.

CA Access Control processes each file record independently. If an error occurs while processing a file, CA Access Control issues a message and continues processing with the next file in the list.

Example: Remove File Protection

The security administrator (which has the ADMIN attribute) wants to remove CA Access Control protection for a file. On UNIX, this can look like this:

```
rmfile /etc/passwd
```

The same command on Windows can look like this:

```
rmfile C:\temp\passwords.txt
```

More information:

chfile Command—Modify File Records (see page 56)
showfile Command—Display File Properties (see page 143)

rm[x]grp Command—Delete Group Records

Valid in the AC environment

The rmgrp and rmxgrp commands remove one or more groups from CA Access Control and, optionally, from the native environment.

Note: There may be occurrences in the database of the group's group ID that the rmgrp command does not delete. For example, the group could be the owner of another group, the owner of other records, or in an access control list for a resource. Use the chgrp, chusr, chres, and authorize commands, as required, to manually change ownership and remove access authorities relating to the group record you want to delete. Alternatively, use the sepurgedb utility to clean up inconsistencies in the database automatically.

Note: The rmgrp command also exists in the native environment but operates differently there.

To use the rmgrp command, at least one of the following is required:

- You have the ADMIN attribute.
- The group to be deleted is within the scope of a group in which you have the GROUP-ADMIN attribute.
- You are the owner of the group to be deleted.
- You are assigned DELETE authority in the GROUP record of the AUDIT class.

This command has the following format:

{rmgrp|rg | rmxgrp|rxg} { *groupName* | (*groupName* [,*groupName*...]) } [unix|nt]

groupName

Specifies the CA Access Control group to be deleted.

nt

(Optional) Deletes a group from the local Windows database in addition to deleting the group from the CA Access Control database.

unix

(Optional) Deletes a group from the local UNIX system in addition to deleting the group from the CA Access Control database.

Example

The user Joe wants to delete the groups DEPT1 and DEPT2 from the database.

- The user Joe has GROUP-ADMIN authority to the SALES group.
- The groups DEPT1 and DEPT2 are owned by the SALES group.

rmxgrp (DEPT1, DEPT2)

More information:

ch[x]grp Command—Change Group Properties (see page 62)
join[x] Command—Add Users to Internal Groups (see page 122)
join[x]- Command—Remove Users from Groups (see page 125)
show[x]grp Command—Display Group Properties (see page 145)
rmgrp Command—Delete UNIX Groups (see page 171)
rmgrp Command—Delete Windows Groups (see page 198)

rmres Command—Delete a Resource

Valid in the AC environment

The rmres command removes resources from the database. Records belonging to the following classes can be deleted using the rmres command: ACVAR, ADMIN, APPL, CATEGORY, CONNECT, FILE, GAPPL, GHOST, GSUDO, GTERMINAL, HNODE, HOST, HOSTNET, HOSTNP, LOGINAPPL, MFTERMINAL, POLICY, PWPOLICY, SECFILE, SECLABEL, SPECIALPGM, SUDO, SURROGATE, TERMINAL, PROGRAM, PROCESS, RULESET, TCP, UACC, and any user defined class.

Note: This command also exists in the native Windows environment but operates differently.

To remove a record from the database, you must meet one of the following conditions:

- You have the ADMIN attribute.
- The resource record is within the scope of a group in which you have the GROUP-ADMIN attribute.
- You are the owner of the resource record.
- You are assigned the DELETE authority in the access control list of the resource class's record in the ADMIN class.

This command has the following format:

{rmres|rr} *className resourceName*

className

Specifies the name of the class to which the resource belongs. To list the resource classes defined to CA Access Control, use the find command. See the find command in this chapter for more information.

resourceName

Specifies the name of the resource record you are deleting. When removing more than one resource, enclose the list of resource names in parentheses and separate the resource names with a space or a comma.

CA Access Control processes each resource record independently. If an error occurs while processing a resource, CA Access Control issues a message and continues processing with the next resource in the list.

Example

The user Admin1 wants to remove the record TERMS from the TERMINAL class in the database.

- The user Admin1 has the ADMIN attribute.

```
rmres TERMINAL TERMS
```

More information:

chres Command—Modify Resource Records (see page 74)
showres Command—Display Resource Properties (see page 147)
rmres Command—Delete a Windows Resource (see page 198)
find Command—List Database Records (see page 113)

rm[x]usr Command—Delete User Records

Valid in the AC environment

The rmusr and rmxusr commands remove users from the CA Access Control database, and also remove references to the user's record that exist in CA Access Control group records.

rmxusr removes an enterprise user from the CA Access Control database. rmusr removes an internal user from the database. The rmusr command can, optionally, remove the user from the native environment as well.

Note: There may be occurrences in the database of the user that rm[x]usr does not delete. For example, the user could be the owner of a group, the owner of other records, or in an access control list for a resource. Use the ch[x]grp, ch[x]usr, ch[x]res, and authorize commands, as required, to manually change ownership and remove access authorities relating to the user record you want to delete. Alternatively, use the sepurgedb utility to clean up inconsistencies in the database automatically.

Note: The rmusr command also exists in the native environment but operates differently there.

To execute the rm[x]usr command you need to meet at least one of the following requirements:

- You have the ADMIN attribute.
- The user record to be deleted is within the scope of a group in which you have the GROUP-ADMIN attribute.
- You are assigned the DELETE authority in the access control list of the USER record in the ADMIN class.
- You are the owner of the user record.

ru is a synonym of rmusr.

rxu is a synonym of rmxusr.

This command has the following format:

{rmusr|ru | rmxusr | rxu} { *userName* | (*userName* [,*userName*...]) } \
[unix|nt] [appl(homedir=yes)]

appl(homedir=yes)

(UNIX only). Deletes the user's home directory

This argument checks for the existence of the user's home directory in /home, /tmp or /users. If the home directory located in another directory, edit the S99DELETE_postrmusrdir.sh script to incorporate it.

Note: You must specify the unix option before you specify this option.

nt

Deletes the user from the Windows environment, in addition to deleting the user from CA Access Control.

Valid only for rmusr.

userName

Defines a user record.

unix

Deletes the user from the UNIX environment, in addition to deleting the user from CA Access Control.

Valid only for rmusr.

Example

The following command deletes enterprise users Terry and Jane from CA Access Control:

```
rxu (Terry, Jane)
```

More information:

ch[x]usr Command—Change User Properties (see page 90)
show[x]usr Command—Display User Properties (see page 149)
rmusr Command—Delete UNIX User (see page 172)
rmusr Command—Delete a Windows User (see page 199)

ruler Command—Select Properties to Display

Valid in AC and native environments

The ruler command defines the ruler for a class, and so lets you define the set of properties of a class that CA Access Control displays.

The ruler command only applies to the hosts of the current session. The properties of each host are displayed in a separate list. If you change hosts, the ruler command does not change the display of properties in the new hosts.

The following users can issue this command:

- Users with the ADMIN, AUDITOR, or OPERATOR attribute.
- Users who have access read in class ADMIN for the class whose ruler they are trying to set. For example, if you have access read in class ADMIN for the record representing class TERMINAL, you can set the ruler for class TERMINAL.

This command has the following format:

ruler *className* [props(all| *propertyName* [,*propertyName*...])]

className

The name of the class whose display you want to change.

[props(all | *propertyName* [,*propertyName*...])]

Specifies the properties to be displayed.

If you omit the props parameter, CA Access Control displays the names of the properties that are in the current ruler.

all

Specifies that all the properties of the class to be displayed.

propName

Specifies a CA Access Control property to be displayed. You can specify up to 40 properties, separated by spaces or commas.

Examples

- The user admin wants CA Access Control to display only two properties for each user: the owner and the user who is notified about changes.

  ruler USER props(NOTIFY, OWNER)

- The user admin wants to display the properties in the current ruler for class USER.

  ruler USER

- The user admin wants CA Access Control to revert to the default ruler, which is to display all the properties in the class USER.

  ruler USER props(all)

More information:

showfile Command—Display File Properties (see page 143)
show[x]grp Command—Display Group Properties (see page 145)
showres Command—Display Resource Properties (see page 147)
show[x]usr Command—Display User Properties (see page 149)

setoptions Command—Set CA Access Control Options

Valid in the AC environment

The setoptions command sets system-wide CA Access Control options in the running system. For example, you can use setoptions to enable or disable security checking for each class, or for all classes, set the password policies, and list the current settings of the CA Access Control options.

Note: This command also exists in the Windows environment, but operates differently there.

You need ADMIN attribute to use the setoptions command, with the exception that you need only AUDITOR or OPERATOR attribute to use the command setoptions list.

This command has the following format:

```
{setoptions|so} \
    [accgrr|accgrr-] \
    [accpacl|accpacl-] \
    [class+ (className)] \
    [class- (className)] \
    [class (className)] \
    [flags{+|-} (l|W)] \
    [cng_adminpwd|cng_adminpwd-] \
    [cng_ownpwd|cng_ownpwd-] \
    [cwarnlist] \
    [dms{+|-}(dms@hostname)] \
    [inactive(nDays)|inactive-] \
    [is_dms{+|-}] \
    [list] \
    [maxlogins(nLogins)|maxlogins-] \
    [password( \
        [{history(nStoredPasswords) | history-}] \
        [(interval(nDays) | interval-)] \
        [(min_life(nDays) | min_life-)] \
        [{rules( \
                [alpha(nCharacters)] \
                [alphanum(nCharacters)] \
                [(bidirectional) | (bidirectional-)] \
                [grace(nLogins)] \
                [lowercase(nCharacters)] \
                [min_len(nCharacters)] \
                [max_len(nCharacters)] \
                [max_rep(nCharacters)] \
                [{namechk|namechk-}] \
                [numeric(nCharacters)] \
                [{oldpwchk|oldpwchk-}] \
                [prohibited(prohibitedCharacters)] \
                [special(nCharacters)] \
                [sub_str_len(nCharacters)] \
                [uppercase(nCharacters)] \
                [use_dbdict|use_dbdict-] \
        )|rules-}] \
    )] \
```

accgrr

Enables the accumulative group rights (ACCGRR) option.

The default value is enabled.

accgrr-

Disables the accumulative group rights (ACCGRR) option.

accpacl

Enables the use of PACLs in all resources.

accpacl-

Disables the use of PACLs.

class (*className*)

Sets or clears a setting for a CA Access Control class.

class+(*className*)

Enables one or more CA Access Control classes. A class must be enabled for CA Access Control to protect resources of that class. A class should be activated only after you have defined the necessary records to allow access to the resources that belong to the class. See the *Endpoint Administration Guide for UNIX* for more information on the resource classes supplied with CA Access Control.

Use one of the following values:

- The name of a CA Access Control class
- SECLEVEL. This enables security level checking.
- PASSWORD. This activates the password rules. On Windows, it also enables arbitrarily long passwords.

class-(*className*)

Disables one or more CA Access Control classes. Resources that belong to a disabled class are not protected by CA Access Control. Use one of the following values:

- The name of a CA Access Control class
- SECLEVEL. This disables security level checking.
- PASSWORD. This disables the password rules. On Windows, this also disables the long passwords.

You cannot disable the classes GROUP, SECFILE, SEOS, UACC, and USER.

cng_adminpwd

Enables users with the PWMANAGER attribute to change the ADMIN user's password.

cng_adminpwd-

Disables users with the PWMANAGER attribute from changing the ADMIN user's password. This is the default setting.

cng_ownpwd

Enables users to change their own passwords through selang.

cng_ownpwd-

Disables users from changing their own passwords through selang. This is the default setting.

cwarnlist

Displays a table with data about which classes are in Warning mode.

dms{+|-}(*dms*@*hostname*)

Adds or removes DMS databases from the list of DMS databases for this database.

flags{+|-} (I|W)

Sets or clears functionality associated with a class. Valid values are:

I

Case-sensitivity for objects in the specified class.

W

Warning mode for the specified class.

Note: Flags are case-sensitive; use uppercase letters.

history(*NStoredPasswords*)

Specifies the number of previous passwords that are stored in a history list. When a password is changed, the previous password is added to the list, and the oldest password is dropped from the list if necessary. CA Access Control prevents a user from changing their password to one that is in the list.

Enter an integer between 1 and 24. If you specify zero, no passwords are saved.

On Windows, the history option enables the use of passwords longer than eight characters. The form of encryption used when storing the password is determined by the setoptions bidirectional or bidirectional- option.

On UNIX, the history option does *not* affect whether long passwords are enabled. Use the passwd_local_encryption_method configuration setting to determine whether long passwords are enabled.

history-

Disables password history checking.

On Windows, this option disables the use of long passwords.

inactive(*nDays*)

Specifies the number of inactive days after which a user's login is suspended. An inactive day is a day when the user does not log in. Enter a positive integer. If inactive is set to zero, the effect is the same as using the inactive- parameter.

inactive-

Disables the inactive login check.

interval(*nDays*)

Sets the number of days that must pass after passwords are set or changed before the system prompts users for a new password. Enter a positive integer or zero. An interval of zero disables password interval checking for users. Set the interval to zero if you do not want passwords to expire.

If the utility segrace is part of the user's login script, CA Access Control informs the users that the current password has expired when the specified number of days is reached. The users can immediately renew the password or continue using the old password until the number of grace logins is reached. After the number of grace logins is reached, the users are denied access to the system and must contact the system administrator to select a new password.

interval-

Cancels the password interval setting.

is_dms+

Designates the current database as a DMS.

is_dms-

Removes the designation of the current database as a DMS.

list

Displays the current CA Access Control settings on the screen.

maxlogins(*nLogins*)

Sets the maximum number of terminals the user can log in to at the same time. A value of 0 (zero) means that the user can log in from any number of terminals concurrently. This value can be overridden by assigning a value in the user's user record.

Note: If maxlogins is set to 1, you cannot run selang. You must shut down CA Access Control, change the maxlogins setting to greater than one, and start CA Access Control again.

maxlogins-

Disables the global maximum logins check. The number of terminals a user can log in is from unlimited, unless the user's login is restricted in the user record of the user.

min_life(*NDays*)

Sets the minimum number of days between password changes. Enter a positive integer.

password

Sets the password options.

rules

Sets one or more password rules that CA Access Control uses to check the quality of new passwords. The rules are:

alpha(*nCharacters*)

Sets the minimum number of alphabetic characters the new password must contain. Enter an integer.

alphanum(*nCharacters*)

Sets the minimum number of alphanumeric characters the new password must contain. Enter an integer.

bidirectional

Specifies that when passwords are sent to other systems as part of PMDB, they are distributed in clear text (within encrypted messages).

On UNIX, this option is equivalent to setting the following passwd section setting value:

Passwd_distribution_encryption_mode=bidirectional

Note: We recommend that you set the configuration setting rather than use the setoptions command.

On Windows, the passwords are stored in the history list with the encryption specified in the registry value:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl\Encryption Package

bidirectional-

Specifies that passwords are sent in their hash encrypted form.

On Windows the hash function used is SHA-1.

On UNIX, this option is equivalent to setting the following passwd section setting value:

Passwd_distribution_encryption_mode=compatibility

Note: We recommend that you set the configuration setting rather than use the setoptions command.

If this option is specified, long passwords cannot be distributed between heterogeneous operating systems.

grace(*nLogins*)

Sets the maximum number of grace logins that are permitted before the user is suspended. The number of grace logins must be between 0 and 255 inclusive.

min_len(*nCharacters*)

Sets the minimum password length. Enter the minimum total number of characters that the new password must contain.

max_len(*nCharacters*)

Sets the maximum password length. Enter the maximum total number of characters that the new password must contain.

lowercase(*nCharacters*)

Sets the minimum number of lowercase characters the new password must contain. Enter an integer.

max_rep(*nCharacters*)

Sets the maximum number of repetitive characters the new password must contain. Enter an integer.

namechk

Checks whether the password contains or is contained by the user's name. By default, CA Access Control performs this check.

namechk-

Turns off the namechk check.

numeric(*nCharacters*)

Sets the minimum number of numeric characters the new password must contain. Enter an integer.

oldpwchk

Checks whether the new password contains or is contained by the password being replaced. By default, CA Access Control performs this check.

oldpwchk-

Turns off the oldpwchk.

prohibited(*prohibitedCharacters*)

Specifies characters a user cannot use in a password. Enter the prohibited characters.

special(*nCharacters*)

Sets the minimum number of special characters the new password must contain. Enter an integer.

sub_str_len(*nCharacters*)

Sets the maximum number of characters the new password can share with the previous password. Enter an integer.

uppercase(*nCharacters*)

Sets the minimum number of uppercase characters the new password must contain. Enter an integer.

use_dbdict | use_dbdict-

Sets the password dictionary. use_dbdict sets the token to **db** and compares passwords against words in the CA Access Control database. use_dbdict- sets the token to **file** and checks passwords against a file specified in the seos.ini file for UNIX or Windows registry for Windows.

rules-

Disables password quality checking. None of the rules specified by the rules argument are used for password quality checking.

Examples: Set CA Access Control Options

- The user John wants to activate the OpsAct class, an installation-defined class used to protect operator actions.

  The user John has the ADMIN attribute.

  ```
  setoptions class+(OpsAct)
  ```

- The user Mike wants to set a password policy that forces users to supply passwords of length at least 6 characters. Mike also wants to activate password policy enforcement.

  The user Mike has the ADMIN attribute.

  ```
  setoptions class+(PASSWORD)
  setoptions password(rules(min_len(6)))
  ```

- The user SecAdmin wants to enable security level checking.

  The user SecAdmin has the ADMIN attribute.

  ```
  setoptions class+(SECLEVEL)
  ```

- The user Janani wants to set a DMS for this database to send notification to.

  The user Janani has the ADMIN attribute.

  ```
  setoptions dms+(apache@myHost)
  ```

Example: Put a Class into Warning Mode

Put a class into Warning mode by setting the Warning property on the class. You can use the setoptions selang command to do this, as follows:

setoptions class(*classname*) flags+ (W)

classname

Defines the name of the class you want to put into Warning mode.

Note: The W flag is case sensitive and must be in uppercase.

To clear Warning mode for the class, you can also use the setoptions command, as follows:

setoptions class(*classname*) flags- (W)

More information:

setoptions Command—Set CA Access Control Windows Options (see page 200)

search Command—List Database Records

Valid in AC and native environments

This is identical to the find command.

More information:

find Command—List Database Records (see page 113)

showfile Command—Display File Properties

Valid in the AC environment

The showfile command lists the properties of a file record. The properties are listed in alphabetical order. CA Access Control processes each record independently and displays information only for those resources for which you have sufficient authority.

Note: This command also exists in the native environment but operates differently there.

To execute a showfile command, at least one of the following conditions is required:

- You have at least one of the following attributes: ADMIN, AUDITOR, and OPERATOR.
- You are the owner of the file.
- You are assigned read authority in the access control list of the object representing the FILE class record in the ADMIN class.
- You have the GROUP-ADMIN or GROUP-AUDITOR attribute in the group that owns the file or that is a parent of the group that owns the file.

This command has the following format:

{showfile|sf} {*fileName* |(*fileName* [,*fileName*...])} \
[addprops(*propName* [,*propName* ...])] \
[next] \
[props(all | *propName* [,*propName* ...])] \
[useprops(*propName* [,*propName* ...])] \
[nt|unix]

addprops(*propName* [*,propName ...*])

Defines properties to be added to the class ruler for this query only.

fileName

Specifies the name of the file record whose properties are to be listed.

CA Access Control processes each file record independently. If an error occurs while processing a file, CA Access Control issues a message and continues processing with the next file in the list.

fileName can contain wildcard characters, and so match multiple file names.

On UNIX, to display the properties of a file whose name contains a special character or space, type an extra slash (/) before the file name.

next

Displays parts of the requested data. This option is useful when the query data is larger than the set query size.

The maximum query size is determined by the query_size configuration setting. The default query_size setting is 100.

nt

Displays the Windows file attributes as well as the CA Access Control properties.

props(all|*propName* [*,propName ...*])

Defines a new ruler for this class for this query and future queries.

unix

Displays the UNIX file attributes as well as the CA Access Control properties.

useprops(*propName* [,*propName* ...])

Defines a ruler for this query only. The class ruler is unaffected.

Example

The user root wants to list the properties of the file record /etc/passwd.

- User root has the ADMIN attribute.

```
showfile /etc/passwd
```

More information:

checklogin Command—Determine Login Information (see page 53)
chfile Command—Modify File Records (see page 56)
rmfile Command—Delete File Records (see page 128)
showfile Command—Display Native File Properties (see page 173)

show[x]grp Command—Display Group Properties

Valid in the AC environment

The show[x]grp command displays the settings of all the CA Access Control properties of a group record. Optionally, the native environment properties are also shown.

Note: The showgrp command also exists in the native environment but operates differently there.

You can execute a show[x]grp command if at least one of the following conditions is true:

- You have at least one of the following attributes: ADMIN, AUDITOR, and OPERATOR.
- You have the GROUP-ADMIN or GROUP-AUDITOR attribute in each group to be listed, or each group to be listed is within the scope of a group in which you have the GROUP-ADMIN attribute.
- You are the owner of the group.
- You are assigned read authority in the access control list of the GROUP record in the ADMIN class.

This command has the following format:

{showgrp|sg} {*groupName* |*groupName* [,*groupName*...])} \
[addprops(*propName*[,*propName* ...])] \
[next] \
[props(all | *propName*[,*propName* ...])] \
[useprops(*propName*[,*propName* ...])] \
[nt|unix]

addprops(*propName* [,*propName* ...])

Defines properties to be added to the ruler for this query only.

groupName

Specifies the group whose properties you want to list.

groupName can contain wildcard characters.

On UNIX, to display the properties of a group whose name contains a special character or space, type an extra slash (/) before the group name.

next

Display parts of the requested data. This option is useful when the query data is larger than the set query size.

The maximum query size is determined by the query_size configuration setting. The default query_size is 100.

nt

Shows the group's details from the local Windows system in addition to the properties in the database.

props(all|*propName* [,*propName* ...])

Defines the ruler for this class for this query and future queries.

useprops(*propName* [,*propName* ...])

Defines a ruler for this query only. The class ruler is unaffected.

unix

Shows the group's details from the local UNIX system in addition to the properties in the database.

Examples

- The user root wants to display the properties of the security group.
  - The user root has the GROUP-ADMIN attribute in the security group.

  showgrp security
- The user admin wants to display the properties of all enterprise groups.
  - The user admin has the ADMIN and AUDITOR attributes.

  showxgrp *

  The properties of all enterprise groups defined to CA Access Control are listed.

More information:

ch[x]grp Command—Change Group Properties (see page 62)
rm[x]grp Command—Delete Group Records (see page 129)
showgrp Command—Display Native Group Properties (see page 174)

showres Command—Display Resource Properties

Valid in the AC environment

The showres command displays the properties of resources belonging to classes in the database. The properties are listed in alphabetical order. The following classes can be listed using the showres command: ACVAR, ADMIN, CATEGORY, CONNECT, FILE, GHOST, GSUDO, GTERMINAL, HOST, HOSTNET, HOSTNP, SECFILE, SECLABEL, SUDO, SURROGATE, TERMINAL, PROGRAM, PROCESS, TCP, UACC, and any user defined class. CA Access Control processes each resource independently and displays information only for those resources for which you have sufficient authority.

Note: This command also exists in the native Windows environment but operates differently there.

showres also displays information about any programs that have become untrusted. The information includes:

- The reason that the program became untrusted.
- The UID of the last user to access the program (not necessarily the user who caused the program to become untrusted).
- The date and time that this user accessed the program.

You can execute a showres command if at least one of the following conditions is true:

- You have at least one of the following attributes: ADMIN, AUDITOR, and OPERATOR.
- You are the owner of the resource.
- You are assigned read authority in the access control list of the object representing the resource class record in the ADMIN class.

This command has the following format:

{showres|sr} *className resourceName* \
[addprops(*propName* [,*propName*...])] \
[next] \
[props(all | *propName* [,*propName*...])] \
[useprops(*propName* [,*propName*...])]

addprops(*propName* [*,propName...*])

Defines properties to be added to the current ruler for this query only.

className

Specifies the name of the class to which the resource belongs. To list the resource classes defined to CA Access Control, use the find command.

next

Displays parts of the requested data. This option is useful when the query data is larger than the set query size.

The maximum query size is determined by the query_size configuration setting. The query size default is set at 100.

props(all|*propName* [*,propName ...*])

Defines a new ruler for this class for this query and future queries.

resourceName

Specifies the name of the resource record whose properties are to be listed. When listing the properties of more than one resource, enclose the list of resource names in parentheses and separate the resource names with a space or a comma.

CA Access Control processes each resource record independently. If an error occurs while processing a resource, CA Access Control issues a message and continues processing with the next resource in the list.

resourceName can contain wildcard characters.

On UNIX, to display the properties of a single resource record whose name contains a special character or space, type an extra slash (/) before the resource name.

useprops(*propName* [,*propName* ...])

Defines a ruler for this query only. The class ruler is unaffected.

Example

The user Admin1 wants to list the properties of the records whose names match the mask ath* in the TERMINAL class.

- User Admin1 has the ADMIN and AUDITOR attributes.

showres TERMINAL ath*

More information:

chres Command—Modify Resource Records (see page 74)
rmres Command—Delete a Resource (see page 130)
showres Command—Display Native Resource Properties (see page 204)
find Command—List Database Records (see page 113)

show[x]usr Command—Display User Properties

Valid in the AC environment

The show[x]usr command displays the values of all the properties of one or more users defined to CA Access Control.

Use showusr to display the properties of internal users. Use showxusr to display the properties of enterprise users.

Note: The showusr command also exists in the native environment but operates differently there.

You can always list the properties of your own user record. To list properties of another user's record, one of the following conditions must be true:

- You are the owner of the user record.
- You have at least one of the following attributes: ADMIN, AUDITOR, and OPERATOR.
- The user record is within the scope of a group in which you have at least one of the following group attributes: ADMIN, AUDITOR, OPERATOR.
- You are assigned read authority in the access control list of the USER record in the ADMIN class.

This command has the following format:

{showusr|su |showxusr |sxu } [{*userName* |(*userName [,userName...*]) }] \
[addprops(*propName* [,*propName...*])] \
[next] \
[props(all | *propName* [,*propName...*])] \
[useprops(*propName*[,*propName...*])] \
[nt|unix]

addprops(*propName* [*,propName...*])

Defines properties to be added to the current ruler for this query only.

next

Displays parts of the requested data. This option is useful when the query data is larger than the set query size.

The maximum query size is determined by the query_size configuration setting. The query size default is set at 100.

nt

Displays the user Windows' properties in addition to the properties in the database.

props(all|*propName* [*,propName ...*])

Defines a new ruler for this class for this query and future queries.

unix

Displays the user's UNIX properties in addition to the properties in the database.

userName

Defines the name of a user. It can include wildcard characters.

On UNIX, to display the properties of a single user record whose name contains a special character or space, type an extra slash (/) before the group name.

If you do not specify *userName*, the command displays the properties of your own user record.

useprops(*propName* [*,propName ...*])

Defines a ruler for this query only. The class ruler is unaffected.

Examples

- The user root wants to list the properties of enterprise user Robin. The root has ADMIN and AUDITOR attributes.

  showxusr Robin

- The user root wants to list the user properties of enterprise users Robin and Leslie. The root has ADMIN and AUDITOR attributes.

  showxusr (Robin,Leslie)

More information:

rm[x]usr Command—Delete User Records (see page 132)
ch[x]usr Command—Change User Properties (see page 90)
showusr Command—Display Native User Properties (see page 175)

source Command—Execute Commands from a File

Valid in all environments

The source command allows you to execute one or more selang commands that have been placed in a file. CA Access Control reads the specified file, executes the commands, and returns a selang prompt. Any user defined in the database can use this command.

This command is like the source command in csh and tcsh in UNIX.

This command has the following format:

source *fileName*

fileName

Specifies the name of the file that contains the selang commands.

Example

The user admin wants to execute the commands in the file called initf1. The user enters the following command:

source initf1

start dbexport Command—Initiate Database Export

Valid in the AC environment

The start dbexport command exports the CA Access Control database of the host you are connected to, and copies the output to a buffer. If you are connected to a PMDB, you can also use this command to export the PMD database.

Note: Use the get dbexport command to view the output.

This command has the following format:

start dbexport [pmdname(*name*)] [filter("*CLASS*, *CLASS*...")] [param("depend=yes")] [param("edit=yes")]

filter("*CLASS*, *CLASS*...")

(Optional) Defines the classes to export from the database. If you do not specify a class, all rules in the database are exported.

param("depend=yes")

(Optional) Specifies to export dependent classes along with the class that you specify in the filter parameter. When you specify this parameter, CA Access Control exports the specified class and the following dependent classes:

- 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.

param("edit=yes")

(Optional) Specifies that CA Access Control changes each rule that creates a new resource or accessor to a rule that modifies the resource or accessor.

Example: If you specify this parameter CA Access Control changes all newres rules to editres rules.

pmdname(*name*)

(Optional) Defines the name of the PMD database to export.

Example: Initiate Database Export

The following example initiates the export of rules that modify FILE and GFILE class resources. The rules are exported from seosdb, the CA Access Control database on the host you are connected to.

```
start dbexport filter("FILE, GFILE")
```

Example: Initiate Database Export with Dependent Classes

The following example initiates the export of rules that modify FILE class resources and any classes that are dependent on FILE class resources, and changes each rule that creates a new resource or accessor to a rule that modifies the resource or accessor:

```
start dbexport filter("FILE") param("depend=yes edit=yes")
```

More information:

get dbexport Command—Retrieve Exported Database Rules (see page 114)

start devcalc Command—Initiate Policy Deviation Calculation

Valid in the AC environment

The start devcalc command initiates policy deviation calculation and sends deviation status. The deviation data is stored in a local policy deviation data file (deviation.dat) and policy deviation status is sent to a DMS through one or more set DHs. To retrieve the actual deviation data, you need to run the get devcalc command.

Note: You do not need to run the deviation calculator manually. If you use advanced policy management, the policyfetcher does this for you regularly. If you have enterprise reporting enabled, the Report Agent also does this regularly. For more information about policy deviation calculation, see the *Enterprise Administration Guide*.

To run the start devcalc command you must have terminal access rights to the computer and execute access to DEVCALC sub-administration class.

This command has the following format:

start devcalc [params("-pn *name#xx* -strict -nonotify -precise")]

-nonotify

(Optional) Specifies that devcalc does *not* send deviation status to the DMS through the DH.

Note: The deviation calculation command policyfetcher runs is defined in the devcalc_command configuration setting and, by default, uses this option to avoid sending deviation status twice.

-pn *name#xx*

(Optional) Defines a comma-separated list of POLICY objects (policy version) the deviation calculator should calculate differences for. If no policy is specified, the deviation calculator calculates differences for all policies deployed on the local host.

-strict

(Optional) Compares between the policies associated with the local HNODE object and the ones associated with the HNODE object on the first available DMS.

Normally, the deviation calculator checks for deviations only on the local host. If this option is specified, the deviation calculator also compares the local policies to the policies on the first available DMS in the list. It compares the:

1. List of policies associated with the HNODE object representing the local host.
2. Policy state of each POLICY object associated with the HNODE object.
3. Policy signature of each POLICY object associated with the HNODE object.

Use this option when you need to validate the result of the deviation calculation.

Note: If you have a large number of endpoints running the deviation calculation simultaneously, the DMS will be heavily loaded. We recommend that you configure your endpoints to use a DMS list or divide your hierarchy into smaller hierarchies and use this option within those smaller hierarchies.

-precise

(Optional) Specifies that the deviation report also displays added objects, properties, and values that exist in the endpoint database and are not found in the policy. By default, the report only displays missing and mismatched items. Use this option when you would like to view the contents on the endpoint database and compare it to the deployed policy.

Example: Start a Policy Deviation Calculation for a Specific Policy

The following example shows how you can use the start devcalc command to calculate policy deviations for the second version of a policy called myPolicy and send the deviation status to the DMS list specified in the local CA Access Control database:

```
AC> start devcalc params("-pn myPolicy#02")
```

start_transaction Command—Start Recording Dual Control Transactions

Valid on UNIX hosts in the AC environment

The start_transaction and end_transaction commands create a file that contains an unprocessed transaction for Dual Control PMDB processes, with one or more commands. The administrator (any user with the ADMIN attribute) who entered the commands in the transaction is called a Maker. The commands must be authorized by a Checker (any administrator who is *not* the Maker) before they are executed in the PMDB.

The Checker must lock transactions before they can be processed. Until the transaction is locked by the Checker, the Maker can retrieve it, change the commands, or delete it. (See the sepmd utility in the *Reference Guide* for details.) When the Maker enters the end_transaction command, the transaction receives a unique id number. If the Maker wants to edit or retrieve the transaction later, this identifying number must be added after the transaction's name in the start_transaction command. When the Maker retrieves the transaction, the name of the Maker, the id number of the transaction, and a short description are displayed (if the Maker entered a description in the *transactionName* parameter).

A Maker cannot change the transactions of other Makers. The objects used in a transaction cannot be used by other Makers in different transactions until the commands have been processed.

Each unprocessed transaction stays in a separate file until a Checker processes it. The Checker can authorize or reject a transaction. If the transaction is authorized, the commands are executed and the PMDB is changed accordingly. If the Checker rejects the transaction, the commands are deleted and the PMDB is not changed.

When the end_transaction command is entered at the end of the Maker's work, the numeric id of the transaction appears. The commands can fail for the following reasons:

- if a command refers to an object that has been used in a different transaction which has not been processed yet
- if a command pertains to the Maker-you cannot change yourself

- if a command contains invalid syntax
- if a command refers to objects that do not exist (in this case a warning message appears)
- You can execute the start_transaction and end_transaction commands if you have the ADMIN attribute.
- Since the hosts command must be executed before invoking the start_transaction and end_transaction commands, you must be authorized to use the hosts command.

Note: For more information on Dual Control, see the *Endpoint Administration Guide for UNIX*.

Usage notes:

- The hosts command must be executed before invoking the start_transaction and end_transaction commands, and the name of the PMDB must be "maker."
- In order for the start_transaction and end_transaction commands to function, the value for the is_maker_checker token in the pmd.ini file and in the [pmd] section of the seos.ini file must be set to yes.

This command has the following format:

start_transaction *transactionName* [*transactionId*]
.
.
.
end_transaction

transactionName

Specifies the name or a description of the transaction. You can enter a string of up to 256 alphanumeric characters.

transactionId

Specifies the unique number given to the transaction when it is created. This numeric id appears automatically when you create a transaction. You must specify this id number when you update the same transaction.

Examples

- The Maker Sally wants to add user Anne to the PMDB, and restrict their access to the system to weekdays between 8:00 a.m. and 8:00 p.m. Then Sally wants to change the default access to the tty30 terminal to read only. Sally wants to call this transaction "general".

  – The Maker has the ADMIN attribute.

  ```
  hosts maker@
  start_transaction general
  newusr anne
  (days(weekdays)time(0800:2000))
  chres TERMINAL tty30
  defaccess(read)
  end_transaction
  ```

  When Sally enters the end_transaction command, the transaction is assigned an ID number, such as seven.

- The Maker Sally wants to add the FINANCIAL category to the user Anne. Sally added the user Anne record earlier the same day, and the command has not yet been processed or implemented on the PMDB.

  – The Maker has the ADMIN attribute.

  ```
  hosts maker@
  start_transaction general 7
  chusr anne category(FINANCIAL)
  end_transaction
  ```

unalias Command—Remove selang Aliases

Valid on UNIX hosts

The unalias command removes an alias defined by the alias command.

Note: You can list all defined aliases and their values using the alias command.

This command has the following format:

unalias *aliasName*

aliasName

Specifies the name of the alias you want to delete from the database.

More information:

alias Command—Define selang Aliases (see page 42)

undeploy Command—Initiate Policy Removal

Valid in the AC environment

This command is a synonym of the deploy- command.

More information:

deploy- Command—Initiate Policy Removal (see page 110)

selang Commands in the Remote Configuration Environment

This section contains a complete alphabetic reference to all the selang commands that operate on the CA Access Control configuration resources (commands in the config environment).

editres config—Modify Configuration Settings

Valid in the config environment

Use the editres config command to modify CA Access Control configuration settings.

The editres config command has different formats for different sets of files. These sets are:

- Audit configuration files (audit.cfg and auditrouteflt.cfg) and PMDB filter files
- All other files

This command has the following syntax for audit configuration files and PMDB filter files:

editres config *name* [line+|-(*value*)] [clear]

This command has the following syntax for all other files:

editres config *name* section(*path*) token[-](*name*) value[+|-](*value*) data_type(*type*)

name

Specifies the configuration resource you want to modify. To modify a PMDB filter file, specify the file name in the format *pmdname*@*filter*, for example, master_pmdb@filter.flt

Note: For a list of configuration resources for the host you are managing, use the *find config* command.

clear

Deletes all values from the audit configuration file or PMDB filter file.

Note: This option does not delete comments from the file.

data_type(*type*)

Specifies the data type of the configuration entry.

Values: str, numeric, multi_str

Default: str

Note: For UNIX, data_type can only be *str*. Other data types are not applicable to UNIX, as it stores configuration settings in files (text strings).

line+(*value*)

Defines the value you want to add to the audit configuration file or PMDB filter file.

Note: The *value* can be a value or a comment.

line-(*value*)

Defines the value you want to remove from the audit configuration file or PMDB filter file.

Note: The *value* can be a value or a comment.

section(*path*)

Defines the section of the configuration resource that you want to modify.

Note: For Windows registry settings, if you do not specify this option, the command modifies the registry key *name* defines.

token(*name*)

Defines the name of the configuration entry that you want to modify.

token-(*name*)

Defines the name of the configuration entry that you want to remove.

value(*value*)

Defines the value that you want to assign to a configuration entry. If a value for the configuration entry already exists, CA Access Control replaces the value with *value*.

If you do not specify a *value*, the command resets the configuration entry value.

value+(*value*)

(Windows REG_MULTI_SZ registry entries only) Defines the value that you want to append to a configuration entry.

(All other configuration values) Defines the value that you want to assign to a configuration entry. If a value for the configuration entry already exists, CA Access Control replaces the value with *value*.

Note: To ensure selang correctly translates the assigned value, enclose the value in quotes (" ").

value-(*value*)

(Windows REG_MULTI_SZ registry entries only) Defines the value that you want to remove from a configuration entry.

(All other configuration values) Specifies to remove any value from the configuration entry.

Examples: Modify ACROOT Configuration Settings on Windows

The following examples show how to modify CA Access Control for Windows configuration settings.

- This example configures CA Access Control to use Audit Only mode:

  ```
  er CONFIG ACROOT section(SeOSD) token(GeneralInterceptionMode) value(1)
  ```

- This example adds a domain name to the list of domain names CA Access Control maintains for host name resolution. The domain_names registry entry is a REG_MULTI_SZ registry entry:

  ```
  er CONFIG ACROOT section(SeOSD) token(domain_names) value+(company.com)
  ```

- This example removes a domain name from the list of domain names CA Access Control maintains for host name resolution. The domain_names registry entry is a REG_MULTI_SZ registry entry:

  ```
  er CONFIG ACROOT section(SeOSD) token(domain_names) value-(company.com)
  ```

- This example removes a configuration setting:

  ```
  er CONFIG ACROOT section(AccessControl) token-(Emulate)
  ```

- This example configures the parent Policy Model of a Policy Model on the managed host:

  ```
  er config myPMDB@PMDROOT token(Parent_Pmd) value(topPMDB@host1.comp.ca)
  ```

Examples: Modify seos.ini Configuration Settings on UNIX

The following examples show how to modify CA Access Control for UNIX configuration settings.

- This example configures CA Access Control to enable PAM authentication:

  er CONFIG seos.ini section(seos) token(pam_enabled) value(yes)

- This example configures the domain name that CA Access Control maintains for host name resolution:

  er CONFIG seos.ini section(seosd) token(domain_names) value+(company.com)

- This example removes the domain name that CA Access Control maintains for host name resolution:

  er CONFIG seos.ini section(seosd) token(domain_names) value-(company.com)

- This example removes a configuration setting:

  er CONFIG seos.ini section(serevu) token-(admin_user)

Example: Modify Audit Configuration File

The following example adds a line to the audit configuration file:

er CONFIG audit.cfg line+("FILE;*;Administrator;*;R;P")

Example: Modify PMD Filter File

The following example adds a line to the PMD filter file:

er config pmdb@filter line+("*;*;USER;*;OLD_PASSWD;PASS")

find config—List Configuration Resources

Valid in the config environment

The find config command lists the CA Access Control configuration resources for the host you are managing. These resources can be registry keys or configuration files.

Possible resources vary by host type:

UNIX	Windows
seos.ini	ACROOT
pmd.ini@*pmd_name*	pmd_name@PMDROOT
	SEOSDRV

This command has the following format:

find config

Note: This command does not return a list of audit.cfg or auditrouteflt.cfg configuration files.

Example: List Configuration Resources for a Windows Host

The following example shows the output of the find config command on a Windows host that has a Policy Model named pmdb:

```
AC(config)> find config
(localhost)
pmdb@PMDROOT
ACROOT
SEOSDRV
```

showres config—Display Configuration Information

Valid in the config environment

Use the showres config command to display CA Access Control configuration information.

The showres config command has different formats for different sets of files. These sets are:

- Audit configuration files (audit.cfg and auditrouteflt.cfg) and PMDB filter files
- All other files

This command has the following syntax for audit configuration files and PMDB filter files:

showres config *name*

This command has the following syntax for all other files:

showres config *name* [section(*path*)] [token(*name*)] [recursive] [sectiononly]

name

Specifies the configuration resource you want to view information about. To view information about a PMDB filter file, specify the file name in the format *pmdname*@*filter*, for example, master_pmdb@filter.flt

Note: For a list of configuration resources for the host you are managing, use the *find config* command.

section(*path*)

(Optional) Defines the section of the configuration resource that you want to view information about.

If you do not specify this option, the command lists all of the configuration entries and sections in the *name* configuration resource.

token(*name*)

(Optional) Defines the name of the configuration entry that you want to view information about.

If you do not specify this option, the command lists all configuration entries and sections in the section(*path*) you defined.

recursive

Specifies to display information about all configuration entries and sections in all sub sections.

sectiononly

Specifies to display information about sections only (no configuration entries will be listed).

selang Commands in the Native UNIX Environment

This section contains a complete alphabetic reference to all the selang commands that operate on the UNIX system files (commands in the native UNIX environment).

chfile Command—Modify UNIX File Settings

Valid in the native UNIX environment

The chfile and editfile commands change the settings of one or more UNIX files.

Note: This command also exists in the AC environment but operates differently.

This command has the following format:

```
{{chfile|cf}|{editfile|ef}} fileName \
    [owner(userName)] \
    [group(groupName)] \
    [mode( \
        [fowner(string)] \
        [fgroup(string)] \
        [fother(string)] \
    )]
```

fileName

Specifies the name of the file whose settings are to be changed. Enter at least one UNIX file name. When changing more than one file, enclose the list of file names in parentheses and separate the file names with a space or a comma.

group(*groupName*)

Changes the group to which the file belongs. Specify a valid group name.

mode

Updates the access modes of the file.

fowner(*string*)

Specifies the access modes for the owner of the file. Use the letters r, w, and x in *string* to assign read, write, and execute permissions, respectively. Use the letter s to make a file setuid.

Specify a plus sign (+) at the beginning of *string* to add permissions to the existing permissions. Specify a minus sign (-) at the beginning of *string* to remove the permissions. If you do not specify a prefix, the previous permissions are reset to *string.*

fgroup(*string*)

Specifies the access modes for the file's group. Use the letters r, w, and x in *string* to assign read, write, and execute permissions, respectively. Use the letter s to make a file setgid.

Specify a plus sign (+) at the beginning of *string* to add permissions to the existing permissions. Specify a minus sign (-) at the beginning of *string* to remove the permissions. If you do not specify a prefix, the previous permissions are reset to *string.*

fother(*string*)

Specifies the access modes that apply to other accessors. Use the letters r, w, and x in *string* to assign read, write, and execute permissions, respectively. Specify a plus sign (+) at the beginning of *string* to add permissions to the existing permissions. Specify a minus sign (-) at the beginning of *string* to remove the permissions. If no prefix is specified, the previous permissions are reset to *string*.

owner(*userName*)

Changes the owner of the file. Specify the user name of a valid UNIX user.

More information:

find file Command—List Native Files (see page 168)
showfile Command—Display Native File Properties (see page 173)
chres Command—Modify Resource Records (see page 74)

chgrp Command—Modify UNIX Groups

Valid in the native UNIX environment

Use the chgrp, editgrp, and newgrp commands to work with UNIX groups. These commands are identical in structure and only vary in the following way:

- The chgrp command *modifies* one or more UNIX groups.
- The editgrp command *creates or modifies* one or more UNIX groups.
- The newgrp command *creates* one or more UNIX groups.

Note: Groups are read, added, updated, and deleted from the file specified in the configuration settings (seos.ini); by default, this file is /etc/group. For more information, see the *Endpoint Administration Guide for UNIX*.

Note: This command also exists in the AC environment but operates differently.

This command has the following format:

{{chgrp|cg}|{editgrp|eg}|{newgrp|ng}} *groupName* \
[groupid(*integer*)] \
[userlist(*userNames*)]

groupid(*integer*)

Sets the group ID of the group. Enter a positive integer representing the group's unique numeric ID. CA Access Control does not allow a group ID of zero.

groupName

Specifies the name of the group to be modified. Specify the name of an existing UNIX group. When altering more than one group, enclose the list of group names in parentheses and separate group names with a space or a comma.

userlist(*userNames*)

Specifies a new member list. Each user name must already be defined to UNIX. When more than one user is in the list, separate the user names with a space or comma. The user list specified here replaces any previous user list defined to the group.

More information:

rmusr Command—Delete UNIX User (see page 172)
showusr Command—Display Native User Properties (see page 175)
ch[x]grp Command—Change Group Properties (see page 62)

chusr Command—Modify UNIX Users

Valid in the native UNIX environment

Use the chgusr, editusr, and newusr commands to work with UNIX users. These commands are identical in structure and vary only in the following ways:

- The chusr command *modifies* one or more UNIX users.
- The editusr command *creates or modifies* one or more UNIX users.
- The newusr command *creates* one or more UNIX users.

Note: Users are read, added, updated, and deleted from the file specified in the configuration settings (seos.ini); by default, this file is /etc/passwd. For more information, see the *Endpoint Administration Guide for UNIX*.

Note: This command also exists in the CA Access Control environment but operates differently there.

This command has the following format:

{{chusr|cu}|{editusr|eu}|{newusr|nu}} *userName* \
[enable] \
[gecos(*string*)] \
[homedir({*path*|nohomedir})] \
[password(*string*)] \
[pgroup(*groupName*)] \
[shellprog(*path*)] \
[userid(*number*)]

enable

Enables the login of a user account that was disabled for any reason. This is a chusr and editusr parameter.

gecos(*string*)

Specifies a string containing general comments about the user, such as the user's full name. Enclose the string in single quotation marks.

homedir(*path*|nohomedir)

Specifies the full path of the user's home directory. CA Access Control attempts to create the directory. The UNIX file is updated, regardless of whether CA Access Control successfully creates the home directory.

If you specify nohomedir, UNIX does not create a homedir for the user.

password(*string*)

Assigns a password to the user. Specify any character except a blank space. The password is valid for one login only. When the user next logs in to the system, a new password must be set.

pgroup(*groupName*)

Specifies the user's primary group name.

shellprog(*path*)

Specifies the full path of the initial program or shell that is executed after the user invokes the login command or the su command.

userid(*number*)

Specifies the user's unique numeric ID, used for unique discretionary access control. Enter a decimal number greater than 100; values less than 100 are not accepted.

userName

The name of an existing UNIX user. When changing more than one user, enclose the list of user names in parentheses and separate the names with a space or a comma.

More information:

rmusr Command—Delete UNIX User (see page 172)
showusr Command—Display Native User Properties (see page 175)
ch[x]usr Command—Change User Properties (see page 90)

editfile Command—Modify UNIX File Settings

Valid in the native UNIX environment

This command is documented with the chfile command.

More information:

chfile Command—Modify UNIX File Settings (see page 163)

editgrp Command—Create and Modify UNIX Groups

Valid in the native UNIX environment

This command is documented with the chgrp command.

More information:

chgrp Command—Modify UNIX Groups (see page 165)

editusr Command—Create and Modify UNIX Users

Valid in the native UNIX environment

This command is documented with the chusr command.

More information:

chusr Command—Modify UNIX Users (see page 166)

find file Command—List Native Files

Valid in the native environment

Use the find file command to list all the system files that match the mask, which is a string. The files are ordered chronologically in one column.

This command has the following format:

find file [*directory*][/*mask*]

directory

Lists all the files in the directory *directory*.

mask

Lists all the files in the directory *directory* that match the *mask* variable. The *mask* may include wildcard characters.

Example: Find Executable Program Files in a Specific Path on Windows

The following command lists all executable files in the CA Access Control bin directory:

find file C:\Program\Files\CA\AccessControl\bin*.exe

Example: Find Files Matching a Pattern on UNIX

The following command lists all files in the CA Access Control bin directory that begin with the letter *se*:

find file /opt/CA/AccessControl/bin/se*

join Command—Add Users to Native Groups

Valid in the native environments

The join command adds users to a group. The specified users and group must already be defined to native OS.

Note: This command also exists in the AC environment but operates differently.

To use the join command, at least one of the following must be true:

- You have the ADMIN attribute in your CA Access Control user record.
- The group record is within the scope of a group in which you have the GROUP-ADMIN attribute.
- You are the owner of the group record in the database.
- You have JOIN or MODIFY access authority in the access control list of the GROUP record in the ADMIN class.

Note: Both the MODIFY and JOIN properties are required if an ADMIN is to have the authority to modify CA Access Control GROUP records and native groups.

This command has the following format:

{join|j} *userName* group(*groupName*)

group(*groupName*)

Specifies the native group to which the users are being added.

userName

Specifies the user name of the native user who is being connected to the group specified by the group parameter. When specifying more than one user, enclose the user names in parentheses and separate the user names with a space or a comma.

Example

The user Eli wants to join the user Bob to the group "staff."

- Eli has the ADMIN attribute and the current environment is *native*.

```
join Bob group(staff)
```

More information:

join- Command—Remove Users from Native Groups (see page 170)
showgrp Command—Display Native Group Properties (see page 174)
showusr Command—Display Native User Properties (see page 175)
join[x] Command—Add Users to Internal Groups (see page 122)

join- Command—Remove Users from Native Groups

Valid in the native environments

The join- command removes users from a group.

Note: This command also exists in the AC environment but operates differently.

To use the join- command, one of the following conditions must be true:

- You have the ADMIN attribute.
- The group record is within the scope of a group in which you have the GROUP-ADMIN attribute.
- You are the owner of the group record in the database.
- You have JOIN or MODIFY access authority in the access control list of the GROUP record in the ADMIN class.

If you only have ownership of the user's profile, you do not have sufficient authority to remove the user from a group. Both the MODIFY and JOIN properties are required if an ADMIN is to have the authority to modify CA Access Control records and native groups

This command has the following format:

{join-|j-} *userName* group(*groupName*)

group(*groupName*)

Specifies the native group from which to remove the user.

userName

Specifies the user name of the user you want to remove from the group. When removing more than one user from the group, enclose the list of user names in parentheses and separate the user names with a space or a comma.

Example

The user Bill wants to remove the users sales25 and sales43 from the PAYROLL group.

- The user Bill has the ADMIN attribute and the current environment is *native*.

```
join- (sales25 sales43) group(PAYROLL)
```

More information:

join Command—Add Users to Native Groups (see page 169)
showgrp Command—Display Native Group Properties (see page 174)
showusr Command—Display Native User Properties (see page 175)
join[x]- Command—Remove Users from Groups (see page 125)

newgrp Command—Create UNIX Groups

Valid in the native UNIX environment

This command is documented with the chgrp command.

More information:

chgrp Command—Modify UNIX Groups (see page 165)

newusr Command—Create UNIX Users

Valid in the native UNIX environment

This command is documented with the chusr command.

More information:

chusr Command—Modify UNIX Users (see page 166)

rmgrp Command—Delete UNIX Groups

Valid in the native UNIX environment

The rmgrp command deletes one or more groups from the UNIX system.

Note: This command also exists in the AC environment but operates differently.

Note: Groups are read, added, updated, and deleted from the file specified in the configuration settings (seos.ini); by default, this file is /etc/group. For more information, see the *Endpoint Administration Guide for UNIX*.

This command has the following format:

{rmgrp|rg} *groupName*

groupName

Specifies the name of the group to be deleted. The group name must be an existing UNIX group name. Specify one or more group names. When removing more than one group, enclose the list of group names in parentheses and separate the group names with a space or a comma.

More information:

chgrp Command—Modify UNIX Groups (see page 165)
showgrp Command—Display Native Group Properties (see page 174)
rm[x]grp Command—Delete Group Records (see page 129)

rmusr Command—Delete UNIX User

Valid in the native UNIX environment

The rmusr command removes one or more users from the UNIX system.

Note: This command also exists in the AC environment but operates differently.

Note: Users are read, added, updated, and deleted from the file specified in the configuration settings (seos.ini); by default, this file is /etc/passwd. For more information, see the *Endpoint Administration Guide for UNIX*.

This command has the following format:

{rmusr|ru} *userName*

userName

Specifies the user name of an existing UNIX user. When removing more than one user, enclose the list of user names in parentheses and separate the user names with a space or a comma.

More information:

chusr Command—Modify UNIX Users (see page 166)
showusr Command—Display Native User Properties (see page 175)
rm[x]usr Command—Delete User Records (see page 132)

showfile Command—Display Native File Properties

Valid in the native environments

The showfile command lists the native details of one or more system files.

Note: This command also exists in the AC environment but operates differently.

This command has the following format:

{showfile|sf} *fileName* [next] \
[{props|addprops}(*propNames*)]

addprops(*propName*)

Sets the properties (ruler) to be displayed. The list of properties is added to the current ruler. The ruler is set for this query only, and reverts to the previously set ruler.

fileName

Specifies the name of the file whose details are to be listed. Enter one or more UNIX file names. When specifying more than one file, enclose the list of file names in parentheses and separate the individual names with a space or a comma.

next

Displays parts of the requested data. This option is useful when the query data is larger than the set query size.

The maximum query size is determined by the query_size configuration setting. The query size default is set at 100.

props(all|*propName*)

Sets the properties (ruler) to be displayed.

The ruler remains set for future queries.

Example: Show the Details of a UNIX File

You want to list the details of the UNIX file /tmp/foo.

```
showfile /tmp/foo
```

Example: Show the Owner of a Windows File

You want to know who the owner of the Windows file C:\tmp\foo.exe is.

```
showfile C:\tmp\foo.exe props(Owner)
```

More information:

chfile Command—Modify UNIX File Settings (see page 163)
chfile Command—Modify Windows File Settings (see page 180)
showfile Command—Display File Properties (see page 143)

showgrp Command—Display Native Group Properties

Valid in the native environments

The showgrp command displays the details of one or more groups in the native operating system.

Note: This command also exists in the AC environment but operates differently.

Note: On UNIX, groups are read, added, updated, and deleted from the file specified in the configuration settings (seos.ini); by default, this file is /etc/group. For more information, see the *Endpoint Administration Guide for UNIX*.

This command has the following format:

{showgrp|sg} *groupName* [next] \
[{props|addprops}(*propNames*)]

addprops(*propName*)

Sets the properties (ruler) to be displayed. The list of properties is added to the current ruler. The ruler is set for this query only, and reverts to the previously set ruler.

groupName

Specifies the name of the group whose details are to be displayed. The group name must be an existing native group name. Specify one or more group names. When listing more than one group, enclose the list of group names in parentheses and separate the group names with a space or a comma.

next

Displays parts of the requested data. This option is useful when the query data is larger than the set query size.

The maximum query size is determined by the query_size configuration setting. The query size default is set at 100.

props(all|*propName*)

Sets the properties (ruler) to be displayed.

The ruler remains set for future queries.

Example

To list details of the UNIX group *security* when you are in the *unix* environment, enter the following command:

```
showgrp security
```

More information:

chgrp Command—Modify UNIX Groups (see page 165)
chgrp Command—Modify Windows Groups (see page 181)
show[x]grp Command—Display Group Properties (see page 145)

showusr Command—Display Native User Properties

Valid in the native UNIX environment

The showusr command displays the properties of one or more users defined in the native operating system.

Note: This command also exists in the AC environment but operates differently.

Note: On UNIX, users are read, added, updated, and deleted from the file specified in the configuration settings (seos.ini); by default, this file is /etc/passwd. For more information, see the *Endpoint Administration Guide for UNIX*.

This command has the following format:

```
{showusr|su} userName [next] \
    [{props|addprops}(propNames)]
```

addprops(*propName*)

Sets the properties (ruler) to be displayed. The list of properties is added to the current ruler. The ruler is set for this query only, and reverts to the previously set ruler.

userName

Specifies the name of the user whose native properties are to be displayed. Specify an existing native user name. When listing the properties of more than one user, enclose the list of user names in parentheses and separate the names with a space or a comma.

next

Displays parts of the requested data. This option is useful when the query data is larger than the set query size.

The maximum query size is determined by the query_size configuration setting. The query size default is set at 100.

props(all|*propName*)

Sets the properties (ruler) to be displayed.

The ruler remains set for future queries.

Example

To list details of the UNIX user *leslie* when you are in the *unix* environment, enter the following command:

```
showusr leslie
```

More information:

chusr Command—Modify UNIX Users (see page 166)
chusr Command—Modify Windows Users (see page 186)
show[x]usr Command—Display User Properties (see page 149)

selang Commands in the Native Windows Environment

This section contains a complete alphabetic reference to all the selang commands that operate on the native Windows environment.

authorize Command—Set Accessors' Authority to Access Windows Resources

Valid in the native Windows environment

The authorize command maintains the lists of users and groups authorized to access a particular resource. Using authorize, you can change a list to:

- Permit access to a resource for specific CA Access Control users or groups.
- Block access to a resource for specific CA Access Control users or groups.
- Change the level of access authority to a resource for specific users or groups.

Note: This command also exists in the AC environment but operates differently.

The following Windows environment classes support ACLs, and can be controlled by the authorize command.

- COM
- DISK
- FILE
- PRINTER
- REGKEY
- SHARE

Classes that do not appear in the list have no access control lists and cannot be controlled by the authorize command.

This command has the following format:

{authorize|auth} *className resourceName* \
[access(*accessValue*)|deniedaccess(*accessvalue*)] \
[gid(*groupName, ...*)] \
[uid(*userName, ...*)]

access(*accessValue*)

Specifies the access authority you want the accessors you identify in the uid or gid parameters to have to the resource.

className

Specifies the name of the class to which *resourceName* belongs.

deniedaccess(*accessvalue*)

Specifies the negative access authority that you want accessors, who you identify in the uid or gid parameters, to have to the resource.

The denied *accessvalue* can be: all, create, delete, join, modify, none, password, or read.

Note: You can only use *accessValue* with the authorize command, not with authorize-.

gid(*groupName*)

Specifies the Windows group or groups whose access authority to the resource you are setting. The value *groupName* represents the name of one or more Windows groups. When specifying more than one group, separate the group names with a space or a comma.

resourceName

The name of the resource record to modify or add. When changing or adding more than one resource, enclose the list of resource names in parentheses and separate the resource names with a space or a comma. At least one resource name must be specified.

CA Access Control processes each resource record independently in accordance with the specified parameters. If an error occurs while processing a resource, CA Access Control issues a message and continues processing with the next resource in the list.

uid(*userName*)

Specifies the Windows users whose access authority to the resource you are setting. *userName* is the user name of one or more Windows users. When specifying more than one user, separate the user names with a space or a comma. To specify all users who are defined in Windows, specify an asterisk (*) for *userName*.

More information:

authorize- Command—Remove Accessors' Authority to Access Windows Resources (see page 178)
chfile Command—Modify Windows File Settings (see page 180)
chgrp Command—Modify Windows Groups (see page 181)
chres Command—Modify Windows Resources (see page 183)
chusr Command—Modify Windows Users (see page 186)
authorize Command—Set Access Authorities on a Resource (see page 44)
Windows Access Authority by Class (see page 31)

authorize- Command—Remove Accessors' Authority to Access Windows Resources

Valid in the native Windows environment

The authorize- command removes the access authority to a resource by deleting the accessors from the standard access control list. This leaves the default access to determine accessors' ability to access a particular resource.

Note: This command also exists in the AC environment but operates differently.

This command has the following format:

{authorize-|auth-} *className resourceName* \
[gid(*groupName, ...*)] \
[uid(*userName, ...*)]

className

Specifies the name of the class to which *resourceName* belongs.

gid(*groupName*)

Specifies the Windows group or groups whose access authority to the resource you are setting. The value *groupName* represents the name of one or more Windows groups. When specifying more than one group, separate the group names with a space or a comma.

resourceName

Specifies the name of the resource record to modify or add. When changing or adding more than one resource, enclose the list of resource names in parentheses and separate the resource names with a space or a comma. At least one resource name must be specified.

CA Access Control processes each resource record independently in accordance with the specified parameters. If an error occurs while processing a resource, CA Access Control issues a message and continues processing with the next resource in the list.

uid(*userName*)

Specifies the Windows users whose access authority to the resource you are setting. *userName* is the user name of one or more Windows users. When specifying more than one user, separate the user names with a space or a comma. To specify all users who are defined in Windows, specify an asterisk (*) for *userName*.

More information:

authorize Command—Set Accessors' Authority to Access Windows Resources (see page 176)
chfile Command—Modify Windows File Settings (see page 180)
chgrp Command—Modify Windows Groups (see page 181)
chres Command—Modify Windows Resources (see page 183)
chusr Command—Modify Windows Users (see page 186)
authorize- Command—Remove Access Authorities from a Resource (see page 49)

chfile Command—Modify Windows File Settings

Valid in the native Windows environment

The chfile and editfile commands are identical. They modify one or more Windows files.

Note: This command also exists in the AC environment but operates differently.

This command has the following format for NTFS file systems:

{{chfile|cf}|{editfile|ef}} *fileName* \
[attrib(*attributeValue*)] \
[attrib(-*attributeValue*)] \
[defaccess(*accessValue*)] \
[owner(*userName*|*groupName*)]

This command has the following format for FAT file systems:

{{chfile|cf}|{editfile|ef}} *fileName* \
[attrib([-]*attributeValue*)]

attrib([-]*attributeValue*)

Specifies a set of attributes that determine the character of the file. When a minus sign (-) precedes the argument *value*, this parameter removes the attribute.

defaccess(*accessValue*)

Specifies the access authority for the Native security built-in group Everyone. All the system users are members of the Everyone group. Providing access to the Everyone group covers all the potential anonymous users in addition to all authenticated users.

Note: Defaccess for an object defined in the CA Access Control environment has a different meaning; the default access authority is the authority granted to any accessor who is not in the resource's CA Access Control list who requests access to the resource. The default access also applies to users not defined in CA Access Control.

The defaccess parameter applies only to NTFS file systems.

owner(*userName*|*groupName*)

Assigns a user or group as the owner of the file record. The owner of the file record has unrestricted access to the file. The owner of the file may always update or delete the file record.

More information:

showfile Command—Display Native File Properties (see page 173)
chfile Command—Modify File Records (see page 56)
Windows File Attributes (see page 475)

chgrp Command—Modify Windows Groups

Valid in the native Windows environment

Use the chgrp, editgrp, and newgrp commands to work with Windows groups. These commands are identical in structure and only vary in the following way:

- The chgrp command *modifies* one or more Windows groups.
- The editgrp command *creates or modifies* one or more Windows groups.
- The newgrp command *creates* one or more Windows groups.

Note: This command also exists in the AC environment but operates differently.

When defining more than one group or changing the properties of more than one group, enclose the list of group names in parentheses and separate the group names with a space or a comma.

Note: To add or remove members from a group use the join or join- command.

This command has the following format:

{{chgrp|cg}|{editgrp|eg}|{newgrp|ng}} *groupName* \
[global] \
[comment(*string*)|comment-] \
[privileges(*privList*)] \
[privileges(-*privList*)] \
[rename_group]

comment(*string*)

Adds an alphanumeric comment string of up to 255 characters to the group record. If you previously added a comment string to the group record, the new string specified here replaces the existing string. If the string contains any blanks, enclose the entire string in single quotation marks.

Standard Windows groups have a descriptive comment added on system installation. If you create a new group in both the Windows and AC environments, CA Access Control inserts the comment "CA Access Control Group."

global

Indicates a global group. Each group name must be unique and cannot currently exist in the Windows database. Windows does not allow groups and users to share the same name.

Note: Use *~groupName* when you create global groups and use the services of CA Access Control version 4.1. Version 4.1 and above support this format for backward compatibility.

groupName

For the command newgrp, specifies the name of the group record added to the database. Each group name must be unique and must not currently exist in the Windows database. Unlike the CA Access Control database, Windows does not allow groups and users to share the same name.

For the command chgrp, specifies the name of the group whose properties you are changing.

When defining more than one group or changing the properties of more than one group, enclose the list of group names in parentheses and separate the group names with a space or a comma.

privileges(*privList*|*-privList*)

Adds specific rights to the Windows group record or, when privList is preceded by a minus sign (-), removes the specified rights. Valid values are any of the privileges available in native Windows.

You can specify this parameter only with the chgrp or editgrp command, and only when you are changing an existing group record. You cannot use it to assign privileges when you are creating a new group record.

rename_group

Renames the group account in the Windows database. All the properties of the old group name apply to the renamed group account. Each group name must be unique and must exist in the Windows database. Unlike the CA Access Control database, Windows does not allow groups and users to share the same name.

Note: When CA Access Control is installed on Windows 2000 with Active Directory, CA Access Control renames the pre-Windows 2000 group name.

chres Command—Modify Windows Resources

Valid in the native Windows environment

Use the chres, editres, and newres commands to work with resource records that belong to a CA Access Control class in the Windows environment. These commands are identical in structure and only vary in the following way:

- The chres command *modifies* one or more resources.
- The editres command *creates or modifies* one or more resources.
- The newres command *creates* one or more resources.

Note: This command also exists in the AC environment but operates differently.

This command has the following formats:

{{chres|cr}|{editres|er}|{newres|nr}} *className resourceName* \
[comment(*string*)|comment-] \
[defaccess(*accessValue*)] \
[dword(*integer*)|string(*string*)|binary(*hexastring*)|multistring(*string*)] \
[location(*string*)|location()] \
[maxusers(*integer*)] \
[owner(*userName*|*groupName*)] \
[share_name(*string*)|sharename-]

or

{{chres|cr}|{editres|er}|{newres|nr}} \
DOMAIN *resourceName* \
[computer(*workstationName*)|computer-(*workstationName*)] \
[domainpwd(*connectPassword*)] \
[trusted(*domainName*)|trusted-(*domainName*)]

binary(*hexastring*)

Specifies the value of a registry key when it is a hexadecimal.

className

Specifies the name of the class to which *resourceName* belongs.

For the newres command, valid values are: REGKEY, REGVAL, OU, and SHARE. For the chres and editres commands, valid values are: COM, DISK, DOMAIN, FILE, PRINTER, REGKEY, REGVAL, SERVICE, DEVICE, SESSION, OU, and SHARE.

comment(*string*)

Adds a comment string to the resource record. If you previously added a comment string to the resource record, the new string specified here replaces the existing string. This parameter is valid for SHARE and PRINTER resources only.

computer(*workstationName*)|computer-(*workstationName*)

Specifies the name of the workstation you are adding to the domain, or, when a minus sign precedes the argument, the name of the workstation you are removing from the domain. This parameter can only be used with DOMAIN resources. You can specify this parameter only with the chres or editres command.

defaccess(*accessValue*)

Specifies the access authority for the Native security built-in group Everyone. All the system users are members of the Everyone group. Providing access to the Everyone group covers all the potential anonymous users in addition to all authenticated users.

Note: Defaccess for an object defined in the CA Access Control environment has a different meaning; the default access authority is the authority granted to any accessor who is not in the resource's CA Access Control list who requests access to the resource. The default access also applies to users not defined in CA Access Control.

The defaccess parameter applies only to NTFS file systems.

domainpwd(*connectPassword*)

Specifies the password an administrator must enter when changing trust relationships.

This parameter can only be used with DOMAIN resources. You can specify this parameter only with the chres or editres command.

dword(*integer*)

Specifies the value of a registry key when it is an integer.

gen_prop(*propertyName*)

Specifies the property for the OU class.

This parameter is valid for the OU class only.

gen_value(*valueName*)

Specifies the property value for the OU class.

This parameter is valid for the OU class only.

location(*string*)

Indicates the location of a printer. Use ( ) with blanks to remove this property.

This parameter is valid for PRINTER resources only.

maxusers(*integer*)

Specifies the maximum number (*integer*) of users that can connect to a shared directory at one time.

This parameter is valid for SHARE resources only.

multistring(*string*)

Specifies the value of a registry key when it is a multistring.

owner(*userName*|*groupName*)

Assigns a user or group as the owner of the resource record. The owner of the resource record has unrestricted access to the resource. The owner of the resource is always permitted to update and delete the resource record. For more information, see the *Endpoint Administration Guide for Windows*.

For FILE or SHARE records on a FAT file system, you may not specify the owner parameter. This parameter is also not valid for DEVICE, DOMAIN, OU, PROCESS, REGVAL, SERVICE, and SESSION resources.

resourceName

The name of the resource record to modify or add. When changing or adding more than one resource, enclose the list of resource names in parentheses and separate the resource names with a space or a comma. At least one resource name must be specified.

CA Access Control processes each resource record independently in accordance with the specified parameters. If an error occurs while processing a resource, CA Access Control issues a message and continues processing with the next resource in the list.

share_name(*shareName*)|share_name-

Identifies the share point for a printer.

This parameter is valid for PRINTER resources only.

string(*string*)

Specifies the value of a registry key when it is a string.

trusted(*domainName*) | trusted-(*domainName*)

Specifies the name of the domain you are adding to trusted domains, or, when a minus sign precedes the argument, the name of the domain you are untrusting. This parameter can only be used with DOMAIN resources. You can specify this parameter only with the chres or editres command.

More information:

rmres Command—Delete a Windows Resource (see page 198)
showres Command—Display Native Resource Properties (see page 204)
chres Command—Modify Resource Records (see page 74)

chusr Command—Modify Windows Users

Valid in the native Windows environment

Use the chgusr, editusr, and newusr commands to work with Windows users. These commands are identical in structure and only vary in the following way:

- The chusr command *modifies* one or more Windows users.
- The editusr command *creates or modifies* one or more Windows users.
- The newusr command *creates* one or more Windows users.

Note: This command also exists in the AC environment but operates differently.

This command has the following format:

{{chusr|cu}|{editusr|eu}|{newusr|nu}} *userName* \
[comment(*string*)|comment-] \
[country(*string*)] \
[expire|expire(*mm/dd/yy*[*@hh:mm*])|expire-] \
[flags{(*accountFlags*)|-(*accountFlags*)}] \
[full_name(*fullName*)] \
[homedir(*homeDir*)] \
[homedrive(*homeDrive*)] \
[location(*string*)] \
[logonserver(*serverName*)] \
[organization(*name*)] \
[org_unit(*name*)] \
[password(*password*)] \
[pgroup(*primaryGroup*)] \
[phone(*string*)] \
[privileges(*privList*)] \
[profile(*path*)] \
[rename_user]
[restrictions(\
days({[mon] [tue] [wed] [thu] [fri] [sat] [sun]}|anyday|weekdays) \
time(*startTime:endTime*|anytime))]\
[restrictions-] \
[resume[(*date*)]|resume-} \
[script(*logonScriptPath*)] \
[suspend[(*date*)] | suspend-] \
[terminals(terminalList)|terminals-(*terminalList*)] \
[workstations(*workstationList*)|workstations-(*workstationList*)|workstations-]

comment(*string*)|comment-

Assigns a comment string to the user record.

The argument is an alphanumeric string of up to 255 characters. If the string contains any blanks, enclose the entire string in single quotation marks.

country(*string*)

Specifies the country where the user is located. This string is not used during the authorization process.

The argument is an alphanumeric string of up to 19 characters. If the string contains any blanks, enclose the entire string in single quotation marks.

expire|expire(*mm/dd/yy[@hh:mm*) | expire-

Sets the date on which the user's account expires. If a date is not specified, the user account expires immediately, provided the user is not currently logged in. If the user is logged in, the account expires when the user logs out.

expire- with the newusr command defines a user account that does not have an expiration date. For the chusr and editusr commands, it removes an expiration date from the specified user account.

The date argument takes the format: *mm*/*dd*/*yy* [@*hh*:*mm*].

flags(*accountFlags*|- *accountFlags*)

Specifies particular attributes of a user's account. See the appendix "Windows Values" for a list of valid flag values.

To remove flags from the user record, precede *accountFlags* with a minus (-).

full_name(*fullName*)

Specifies the full name of the user associated with the user record.

The argument is an alphanumeric string of up to 256 characters. If the string contains any blanks, enclose the entire string in single quotation marks.

gecos(*string*)

Specifies a comment string for the user, such as the user's full name. Enclose the string in single quotation marks.

homedir(*homeDir*)

Specifies the user's home directory. Users log in automatically to their own home drives and home directories.

homedrive(*homeDrive*)

Specifies the drive of the user's home directory. Users log in automatically to their own home drives and home directories.

location(*string*)

Specifies the user's location. This string is not used during the authorization process.

The argument is an alphanumeric string of up to 19 characters. If the string contains any blanks, enclose the entire string in single quotation marks.

logonserver(*serverName*)

Specifies the server that verifies the login information for the user. When the user logs in to the domain workstation, CA Access Control transfers the login information to the server, which gives the workstation permission for the user to work.

organization(*name*)

Specifies the organization in which the user works. This information is not used during the authorization process.

The argument is an alphanumeric string of up to 256 characters. If the string contains any blanks, enclose the entire string in single quotation marks.

org_unit(*name*)

Specifies the organizational unit in which the user works. This information is not used during the authorization process.

The argument is an alphanumeric string of up to 256 characters. If the string contains any blanks, enclose the entire string in single quotation marks.

password(*password*)

Assigns a password to a user. If password checking is enabled, the password is valid for one login only. When the user next logs in to the system, a new password must be set.

The argument is a string of up to 14 characters, and cannot include either a space or a comma. If password checking is enabled, the password is valid for one login only. When the user next logs in to the system, the user must set a new password, unless you set the flag for "Password Never Expires".

To change your own password, you need to set selang options using *setoptions cng_ownpwd* or use sepass.

If you are setting passwords for users on Windows NT systems, the following message may appear:

The password is shorter than required.

This error means that the password does not meet the policy requirements. This is caused by any of the following:

- The password is shorter or longer than the required length.
- The password has been used recently and exists in the Windows NT Change History field.
- The password does not have enough unique characters.
- The password does not meet other password policy requirements (such as those set with CA Access Control password policies).

To avoid this error, make sure you set a password which meets all applicable requirements.

pgroup(*primaryGroup*)

Sets the user's primary group ID. A primary group is one of the groups in which a user is defined and must be a Global group.

The argument is a string of up to 14 characters, and cannot include either a space or a comma.

phone(*string*)

Specifies the user's phone number. This information is not used during the authorization process.

privileges(*privList*)

Adds specific rights to the Windows user record or, when *privList* is preceded by a minus sign (-), removes the specified rights. You can specify this parameter only with the chusr or editusr command, and only when you are changing an existing user record. You cannot use it to assign privileges when you are creating a new user record.

profile(*path*)

Specifies the full path location of the file that contains a user's profile for the Desktop environment (program groups, network connections). Every time the user logs in to any workstation, the same environment appears on the screen.

rename_user

Renames the user account in the Windows database. All the properties of the old user name apply to the renamed user account. Each user name must be unique and must exist in the Windows database. Unlike the CA Access Control database, Windows does not allow groups and users to share the same name.

Note: When CA Access Control is installed on Windows 2000 with Active Directory, CA Access Control renames the user logon name (the pre-Windows 2000 user name). However, CA Access Control does not rename the full name as Windows does.

Note: The maximum length of an object name is 255 characters. CA Access Control and Windows do not manage resources with names exceeding 255 characters.

restrictions([*days*] [*time*])|restrictions-([*days*] [*time*])

Specifies the days of the week and the hours in the day when users may access the file.

If you omit the days argument and specify the time argument, the time restriction applies to any day-of-week restriction already indicated in the record. If you omit time and specify days, the day restriction applies to any time restriction already indicated in the record. If you specify both days and time, the users may access the system only during the specified time period on the specified days.

- [Days] specifies the days on which users may access the file. The days argument takes the following sub-arguments:
  - **anyday**-Allow users access to the file on any day.

- **weekdays**-Allow users access to the resource only on weekdays-Monday through Friday.
  - **Mon**, **Tue**, **Wed**, **Thu**, **Fri**, **Sat**, **Sun**-Allow users access to the resource only on the specified days. You can specify the days in any order. If you specify more than one day, separate the days with a space or a comma.
- [Time] specifies the period during which users may access the resource. The time argument takes the following sub-arguments:
  - **anytime**-Allow users access to the resource at any time of the day.
  - **startTime:endTime**-Allow access to the resource only during the specified period. The format of both startTime and endTime is *hhmm*, where *hh* is the hour in 24-hour notation (00 through 23) and *mm* is the minutes (00 through 59). Note that 2400 is not a valid time value. startTime must be less than endTime, and both times must occur on the same day. If the terminal is in a different time zone from the processor, adjust the time values by translating the start and end times for the terminal to the equivalent local times for the processor. For example, if the processor is in New York and the terminal is in Los Angeles, to allow access to the terminal from 8:00 a.m. to 5:00 p.m. in Los Angeles, specify time (1100:2000).

resume(*date*)|resume-

The date, and optionally time, at which Windows will reinstate the user account. If you specify both the suspend parameter and the resume parameter, make sure the resume date falls after the suspend date or the user will stay suspended indefinitely.

Enter a date, and optional time, in the following format:

mm/dd/yy[@HH:MM]

Use resume- parameter to change the status of the user account from active (enabled) to suspended. Use this parameter with the chusr or editusr commands only.

script(*loginScriptPath*)

Specifies the location of a file that runs automatically when the user logs in. This login script configures the working environment. This parameter is optional, since the profile parameter also sets up the user's working environment.

suspend(*date*)|suspend-

Disables a user account. A user cannot use a suspended user account to log in to the system. If you specify date, Windows suspends the user account on the specified date. If you omit a date, Windows suspends the user account immediately upon execution of the chusr command.

Enter a date, and optional time, in the following format: *mm/dd/yy[@HH:MM]*.

Use the suspend- parameter to change the status of the user account from disabled to active (enabled). Use this parameter with the chusr or editusr commands only.

terminals(*terminalList*)|terminals-(*terminalList*)

Specifies up to eight terminals from which the user can log in. Surround the list with quotation marks, and separate the names with commas. For example:

"terminal1,terminal2"

workstations(*workstationList*)|workstations-(*workstationList*)|workstations-

Specifies up to eight workstations from which the user can log in. Surround the list with quotation marks, and separate the names with commas. For example:

"workstation1,workstation2"

editfile Command—Modify Windows File Settings

Valid in the native Windows environment

This command is documented with the chfile command.

More information:

chfile Command—Modify Windows File Settings (see page 180)

editgrp Command—Create and Modify Windows Groups

Valid in the native Windows environment

This command is documented with the chgrp command.

More information:

chgrp Command—Modify Windows Groups (see page 181)

editusr Command—Create and Modify Windows Users

Valid in the native Windows environment

This command is documented with the chusr command.

More information:

chusr Command—Modify Windows Users (see page 186)

editres Command—Create and Modify Windows Resources

Valid in the native Windows environment

This command is documented with the chres command.

More information:

chres Command—Modify Windows Resources (see page 183)

find file Command—List Native Files

Valid in the native environment

Use the find file command to list all the system files that match the mask, which is a string. The files are ordered chronologically in one column.

This command has the following format:

find file [*directory*][/*mask*]

directory

Lists all the files in the directory *directory*.

mask

Lists all the files in the directory *directory* that match the *mask* variable. The *mask* may include wildcard characters.

Example: Find Executable Program Files in a Specific Path on Windows

The following command lists all executable files in the CA Access Control bin directory:

find file C:\Program\Files\CA\AccessControl\bin*.exe

Example: Find Files Matching a Pattern on UNIX

The following command lists all files in the CA Access Control bin directory that begin with the letter *se*:

```
find file /opt/CA/AccessControl/bin/se*
```

find {xuser|xgroup} Command—List Enterprise Users or Groups

Valid in the native Windows environment

The find {xuser|xgroup} command lists the names of enterprise users or groups in the current or *trusted* domains.

Note: This command is supported only on supported Windows 2000 operating systems with Directory Services.

This command has the following format:

```
find {xuser|xgroup} mask [domain(domainName)] [next]
```

xgroup

Specifies for the command to return enterprise groups.

xuser

Specifies for the command to return enterprise users.

domain(*domainName*)

Defines the trusted domain to restrict the search to.

If you do not specify this option, the command returns users from the current domain.

mask

Defines a mask for the enterprise users.

next

Specifies that selang output should continue the listing of enterprise users or groups that was started by a previous find xuser or find xgroup command.

Use this option if there are more than 100 items in the list.

Example: Display Enterprise Users

The following command lists the first 100 enterprise users in the current domain that begin with abc:

```
find xuser abc*
```

join Command—Add Users to Native Groups

Valid in the native environments

The join command adds users to a group. The specified users and group must already be defined to native OS.

Note: This command also exists in the AC environment but operates differently.

To use the join command, at least one of the following must be true:

- You have the ADMIN attribute in your CA Access Control user record.
- The group record is within the scope of a group in which you have the GROUP-ADMIN attribute.
- You are the owner of the group record in the database.
- You have JOIN or MODIFY access authority in the access control list of the GROUP record in the ADMIN class.

Note: Both the MODIFY and JOIN properties are required if an ADMIN is to have the authority to modify CA Access Control GROUP records and native groups.

This command has the following format:

{join|j} *userName* group(*groupName*)

group(*groupName*)

Specifies the native group to which the users are being added.

userName

Specifies the user name of the native user who is being connected to the group specified by the group parameter. When specifying more than one user, enclose the user names in parentheses and separate the user names with a space or a comma.

Example

The user Eli wants to join the user Bob to the group "staff."

- Eli has the ADMIN attribute and the current environment is *native*.

```
join Bob group(staff)
```

More information:

join- Command—Remove Users from Native Groups (see page 170)
showgrp Command—Display Native Group Properties (see page 174)
showusr Command—Display Native User Properties (see page 175)
join[x] Command—Add Users to Internal Groups (see page 122)

join- Command—Remove Users from Native Groups

Valid in the native environments

The join- command removes users from a group.

Note: This command also exists in the AC environment but operates differently.

To use the join- command, one of the following conditions must be true:

- You have the ADMIN attribute.
- The group record is within the scope of a group in which you have the GROUP-ADMIN attribute.
- You are the owner of the group record in the database.
- You have JOIN or MODIFY access authority in the access control list of the GROUP record in the ADMIN class.

If you only have ownership of the user's profile, you do not have sufficient authority to remove the user from a group. Both the MODIFY and JOIN properties are required if an ADMIN is to have the authority to modify CA Access Control records and native groups

This command has the following format:

{join-|j-} *userName* group(*groupName*)

group(*groupName*)

Specifies the native group from which to remove the user.

userName

Specifies the user name of the user you want to remove from the group. When removing more than one user from the group, enclose the list of user names in parentheses and separate the user names with a space or a comma.

Example

The user Bill wants to remove the users sales25 and sales43 from the PAYROLL group.

- The user Bill has the ADMIN attribute and the current environment is *native*.

```
join- (sales25 sales43) group(PAYROLL)
```

More information:

join Command—Add Users to Native Groups (see page 169)
showgrp Command—Display Native Group Properties (see page 174)
showusr Command—Display Native User Properties (see page 175)
join[x]- Command—Remove Users from Groups (see page 125)

newgrp Command—Create Windows Groups

Valid in the native Windows environment

This command is documented with the chgrp command.

More information:

chgrp Command—Modify Windows Groups (see page 181)

newres Command—Create Windows Resources

Valid in the native Windows environment

This command is documented with the chres command.

More information:

chres Command—Modify Windows Resources (see page 183)

newusr Command—Create Windows Users

Valid in the native Windows environment

This command is documented with the chusr command.

More information:

chusr Command—Modify Windows Users (see page 186)

rmgrp Command—Delete Windows Groups

Valid in the native Windows environment

The rmgrp command deletes one or more groups from the Windows database.

Note: This command also exists in the AC environment but operates differently.

This command has the following format:

{rmgrp|rg} *groupName*

groupName

Specifies the name of the group to be deleted. The group name must be an existing Windows group name. Specify one or more group names. When removing more than one group, enclose the list of group names in parentheses and separate the group names with a space or a comma.

rmres Command—Delete a Windows Resource

The rmres command removes one or more resources from the Windows system database.

Note: This command also exists in the AC environment but operates differently.

This command has the following format:

{rmres|rr} *className resourceName*

className

Specifies the name of the class the resource belongs to.

resourceName

Specifies the name of an existing Windows resource of class *className*. When removing more than one resource, enclose the list of user names in parentheses and separate the names with a space or a comma.

More information:

chres Command—Modify Windows Resources (see page 183)
showres Command—Display Native Resource Properties (see page 204)
rm[x]usr Command—Delete User Records (see page 132)

rmusr Command—Delete a Windows User

Valid in the native Windows environment

The rmusr command removes one or more users from the Windows system database.

Note: This command also exists in the AC environment but operates differently.

This command has the following format:

{rmusr|ru} *userName*

userName

Specifies the user name of an existing Windows user. When removing more than one user, enclose the list of user names in parentheses and separate the user names with a space or a comma.

More information:

chusr Command—Modify Windows Users (see page 186)
showusr Command—Display Native User Properties (see page 175)
rm[x]usr Command—Delete User Records (see page 132)

setoptions Command—Set CA Access Control Windows Options

The setoptions command dynamically sets system-wide CA Access Control options related to the Windows operating system.

Note: This command also exists in the AC environment, but operates differently there.

You need ADMIN attribute to use the setoptions command, with the exception that you need only AUDITOR or OPERATOR attribute to use the command setoptions list.

This command has the following format:

```
setoptions|so \
    [audit_policy( \
        [success(system|logon|access|rights \
                |process|security|manage)] \
        [failure(system|logon|access|rights \
                |process|security|manage)] \
    )]
    [password(
        [history(number-stored-passwords)]
        [interval(nDays)]
        [min_life(NDays)]
        )]
```

audit_policy{+|-}

Specifies whether auditing is enabled (+) or disabled (-).

audit_policy(success(system|logon|access|rights|process|security|manage))

Specifies which detected authorized access events are logged. The types of access are:

- **system**-attempts to shutdown or restart the computer.
- **logon**-attempts to log on to or log off from the system.
- **access**-attempts to access securable objects, such as files.
- **rights**-attempts to use Windows Server privileges.
- **process**-events such as program activation, some forms of handle duplication, indirect access to an object, and process exit.
- **security**-attempts to change Policy object rules.
- **manage**-attempts to create, delete, or change user or group accounts. Also, password changes.

audit_policy(failure(system|logon|access|rights|process|security|manage))

Specifies which detected unauthorized access events are logged. The types of access are:

- **system**-attempts to shutdown or restart the computer.
- **logon**-attempts to log on to or log off from the system.
- **access**-attempts to access securable objects, such as files.
- **rights**-attempts to use Windows Server privileges.
- **process**-events such as program activation, some forms of handle duplication, indirect access to an object, and process exit.
- **security**-attempts to change Policy object rules.
- **manage**-attempts to create, delete, or change user or group accounts. Also, password changes.

history(*number-stored-passwords*)

Specifies the number of previous passwords that are stored in the database. When supplying a new password, the user cannot specify any of the passwords stored in the history list. *NStoredPasswords* is an integer between 1 and 24. If you specify zero, no passwords are saved.

interval(*nDays*)

Sets the number of days that must pass after passwords are set or changed before the system prompts users for a new password.

The value of *nDays* must be a positive integer or zero. An interval of zero disables password interval checking for users. Set the interval to zero if you do not want passwords to expire.

min_life(*NDays*)

Sets the minimum number of days between password changes. *NDays* must be a positive integer.

More information:

showfile Command—Display File Properties (see page 143)

showfile Command—Display Native File Properties

Valid in the native environments

The showfile command lists the native details of one or more system files.

Note: This command also exists in the AC environment but operates differently.

This command has the following format:

{showfile|sf} *fileName* [next] \
[{props|addprops}(*propNames*)]

addprops(*propName*)

Sets the properties (ruler) to be displayed. The list of properties is added to the current ruler. The ruler is set for this query only, and reverts to the previously set ruler.

fileName

Specifies the name of the file whose details are to be listed. Enter one or more UNIX file names. When specifying more than one file, enclose the list of file names in parentheses and separate the individual names with a space or a comma.

next

Displays parts of the requested data. This option is useful when the query data is larger than the set query size.

The maximum query size is determined by the query_size configuration setting. The query size default is set at 100.

props(all|*propName*)

Sets the properties (ruler) to be displayed.

The ruler remains set for future queries.

Example: Show the Details of a UNIX File

You want to list the details of the UNIX file /tmp/foo.

```
showfile /tmp/foo
```

Example: Show the Owner of a Windows File

You want to know who the owner of the Windows file C:\tmp\foo.exe is.

```
showfile C:\tmp\foo.exe props(Owner)
```

More information:

chfile Command—Modify UNIX File Settings (see page 163)
chfile Command—Modify Windows File Settings (see page 180)
showfile Command—Display File Properties (see page 143)

showgrp Command—Display Native Group Properties

Valid in the native environments

The showgrp command displays the details of one or more groups in the native operating system.

Note: This command also exists in the AC environment but operates differently.

Note: On UNIX, groups are read, added, updated, and deleted from the file specified in the configuration settings (seos.ini); by default, this file is /etc/group. For more information, see the *Endpoint Administration Guide for UNIX*.

This command has the following format:

{showgrp|sg} *groupName* [next] \
[{props|addprops}(*propNames*)]

addprops(*propName*)

Sets the properties (ruler) to be displayed. The list of properties is added to the current ruler. The ruler is set for this query only, and reverts to the previously set ruler.

groupName

Specifies the name of the group whose details are to be displayed. The group name must be an existing native group name. Specify one or more group names. When listing more than one group, enclose the list of group names in parentheses and separate the group names with a space or a comma.

next

Displays parts of the requested data. This option is useful when the query data is larger than the set query size.

The maximum query size is determined by the query_size configuration setting. The query size default is set at 100.

props(all|*propName*)

Sets the properties (ruler) to be displayed.

The ruler remains set for future queries.

Example

To list details of the UNIX group *security* when you are in the *unix* environment, enter the following command:

```
showgrp security
```

More information:

chgrp Command—Modify UNIX Groups (see page 165)
chgrp Command—Modify Windows Groups (see page 181)
show[x]grp Command—Display Group Properties (see page 145)

showres Command—Display Native Resource Properties

Displays the properties of Windows resources.

This command has the following format:

```
showres|sr className resourceName [next] \
    [{props|addprops}(propNames)]
```

addprops(*propName*)

Sets the properties (ruler) to be displayed. The list of properties is added to the current ruler. The ruler is set for this query only, and reverts to the previously set ruler.

className

Specifies the name of the class the resource belongs to.

next

Displays parts of the requested data. This option is useful when the query data is larger than the set query size.

The maximum query size is determined by the query_size configuration setting. The query size default is set at 100.

props(all|*propName*)

Sets the properties (ruler) to be displayed.

The ruler remains set for future queries.

resourceName

Specifies the name of an existing Windows resource of class *className*.

showusr Command—Display Native User Properties

Valid in the native UNIX environment

The showusr command displays the properties of one or more users defined in the native operating system.

Note: This command also exists in the AC environment but operates differently.

Note: On UNIX, users are read, added, updated, and deleted from the file specified in the configuration settings (seos.ini); by default, this file is /etc/passwd. For more information, see the *Endpoint Administration Guide for UNIX*.

This command has the following format:

{showusr|su} *userName* [next] \
[{props|addprops}(*propNames*)]

addprops(*propName*)

Sets the properties (ruler) to be displayed. The list of properties is added to the current ruler. The ruler is set for this query only, and reverts to the previously set ruler.

userName

Specifies the name of the user whose native properties are to be displayed. Specify an existing native user name. When listing the properties of more than one user, enclose the list of user names in parentheses and separate the names with a space or a comma.

next

Displays parts of the requested data. This option is useful when the query data is larger than the set query size.

The maximum query size is determined by the query_size configuration setting. The query size default is set at 100.

props(all|*propName*)

Sets the properties (ruler) to be displayed.

The ruler remains set for future queries.

Example

To list details of the UNIX user *leslie* when you are in the *unix* environment, enter the following command:

```
showusr leslie
```

More information:

chusr Command—Modify UNIX Users (see page 166)
chusr Command—Modify Windows Users (see page 186)
show[x]usr Command—Display User Properties (see page 149)

xaudit Command—Modify System Access Control List

The xaudit command adds entries in the system access control list (SACL). Each entry in this list causes an audit message to be logged when a specified user or group attempts to gain access to the resource. The xaudit- command removes entries from the SACL, and is valid for resource types FILE, PRINTER, REGKEY, DISK, COM, or SHARE.

This command has the following format:

```
xaudit className resourceName \
    [failure(auditMode)]          \
    [gid(groupName)] \
    [success(auditMode)] \
    [uid(userName)]
```

className

Specifies the name of the resource type to which the resource belongs.

failure(*auditMode*)

Logs unauthorized access attempts to the resource.

Valid values for *auditmode* depend on the resource type to which it belongs:

Note: Only NTFS files can have audit modes

- **DISK** and **COM**: changePermissions, delete, modify, query, read, synchronize, takeOwnership.
- **FILE**: changePermissions, delete, execute, read, takeOwnership, and write.
- **PRINTER**: changePermissions, delete, print, and takeOwnership.
- **REGKEY**: delete, enumerate, link, notify, queryValue, readControl, setValue, subkey, and write.

For all resource types: *none* and *all*.

gid(*groupName*)

Specifies the groups whose access to the resource is being audited. When specifying more than one group, separate the names with spaces or commas.

resourceName

Specifies the name of the resource record whose system access control list (SACL) is being modified.

success(*auditMode*)

Logs authorized accesses to the resource.

Valid values for *auditmode* depend on the resource type to which it belongs:

Note: Only NTFS files can have audit modes

- **DISK** and **COM**: changepermissions, delete, modify, query, read, synchronize, takeownership.
- **FILE**: changePermissions, delete, execute, read, takeOwnership, and write.
- **PRINTER**: changePermissions, delete, print, and takeOwnership.
- **REGKEY**: delete, enumerate, link, notify, queryValue, readControl, setValue, subkey, and write.

For all resource types: *none* and *all*.

uid(*userName*)

Specifies the user whose access to the resource is being audited. When specifying more than one user, separate the user names with spaces or commas. To specify all users who are defined in the Windows NT database, specify an asterisk (*) for *userName*.

More information:

xaudit- Command—Remove System Access Control List (see page 207)

xaudit- Command—Remove System Access Control List

The xaudit- command removes entries from the SACL, and is valid for resource types FILE, PRINTER, REGKEY, DISK, COM, or SHARE.

This command has the following format:

```
xaudit- className, resourceName \
   [gid(groupName)] \
   [uid(userName)]
```

className

Specifies the name of the resource type to which the resource belongs.

gid(*groupName*)

Specifies the groups or groups whose access to the resource is being audited. When specifying more than one group, separate the names with spaces or commas.

resourceName

Specifies the name of the resource record whose system access control list (SACL) is being removed.

uid(*userName*)

Specifies the user whose access to the resource is being audited. When specifying more than one user, separate the user names with spaces or commas. To specify all users who are defined in the Windows NT database, specify an asterisk (*) for *userName*.

More information:

xaudit Command—Modify System Access Control List (see page 206)

selang Commands in the Policy Model Environment

This section contains a complete alphabetic reference to all the selang commands that operate on the Policy Model environment.

backuppmd Command—Back up a PMDB

Valid in the pmd environment

The backuppmd command backs up the data in the PMDB database to a specified directory. All the data in the PMDB database is backed up, including policies, deployment information, and configuration files.

This command has the following format for DMSs:

backup *pmdName* destination(*path*)

This command has the following format for PMDBs:

backup *pmdName* [destination(*path*)|hir_host(*name*)]

destination(*path*)

Defines the directory that you want the backup files to be stored in.

Note: If you do not specify a path, the files will be backed up to the default location specified in the *_pmd_backup_directory_* token.

Default: (UNIX) *ACInstallDir*/data/policies_backup/pmdName

Default: (Windows) *ACInstallDir*\data\policies_backup\pmdName

pmdName

Defines the name of the PMDB or DMS to back up.

hir_host(*name*)

Backs up all the PMDBs in a hierarchy to the host *name* that you specify, and modifies the PMDB subscribers so that the subscription still works when the backup is moved to the *name* host.

Note: This command is only supported if the master and child PMDBs are deployed on the same host.

createpmd Command—Create a PMDB on a Host

Valid in the pmd environment

The createpmd command defines a PMDB on a remote host. You can designate one or more users as administrator, auditor, and password managers for the PMDB. You can also define the PMDB's parent and subscriber PMDBs. You can run createpmd command from a remote host.

This command has the following format:

createpmd *pmdname* \
[admins(*user [user ...]*)] \
[auditors(*user [user ...]*)] \
[pwman(*user [user ...]*)] \
[parentpmd(*pmdname@host*)] \
[desktop(*host-names...*)] \
[subscriber(*host-names*|*pmdnames...*)] \
[pwdfile(*file-name*))] \
[grpfile(*file-name*))] \
[nis] \
[xadmins(*user [user ...]*)] \
[xauditors(*user [user ...]*)] \

admins(*user [user ...]*)

Specifies one or more internal users to be PMDB administrators. Separate multiple users with spaces.

auditors(*user [user ...]*)

Specifies one or more internal users who can view the audit file of the PMDB. Separate multiple users with spaces.

pwmans(*user [user ...]*)

Specifies one or more users as PMDB password managers. Separate multiple users with spaces.

parentpmd(*pmdname*@*host*)

Specifies the name of the PMDB that is a parent to the one you are creating.

Note: If you want to define multiple parent Policy Models with a selang remote command, you must to use quotation marks. For example, to create a Policy Model and define its parent, use the following command:

```
createpmd subs2 admins(abc123 root) auditors(abc123 root) desktop(pcp36949) \
parentpmd("aa@pcp36949,bb@pcp36949")
```

desktop(*host [host ...]*)

Specifies one or more hosts from which administrators can administer the PMDB. Separate multiple hosts with spaces. The default is the host of the new PMDB.

subscribers(*host | pmd [host | pmd ...]*)

Specifies the hosts or PMDBs to be a subscriber of the new PMDB. Separate multiple hosts or pmds with spaces.

pwdfile(*filename*)

Specifies the PMDB password file.

grpfile(*filename*)

Specifies the PMDB group file.

nis

Performs an NIS setup on the new PMDB's host, and creates a filter file to filter out all UNIX updates.

xadmins(*user [user ...]*)

Specifies one or more enterprise users to be PMDB administrators. Separate multiple users with spaces.

xauditors(*user [user ...]*)

Specifies one or more enterprise users who can view the audit file of the PMDB. Separate multiple users with spaces.

pwmans(*user [user ...]*)

Specifies one or more enterprise users as PMDB password managers. Separate multiple users with spaces.

deletepmd Command—Remove a PMDB from a Host

Valid in the pmd environment

The deletepmd command removes the following items from the host:

- The PMDB's selang protection files:
  - database files
  - registry entries
- The contents of the PMDB directory
- The PMDB directory

Important! To prevent serious operational problems, avoid removing the PMDB by manually deleting its files. Always use the deletepmd command for PMDBs.

This command has the following format:

deletepmd *pmdname*

findpmd Command—List PMDBs on the Host

Valid in the pmd environment

The findpmd command lists the PMDBs in the host to which you are connected and whether their daemons are loaded.

This command has the following format:

```
findpmd
```

listpmd Command—List Information about a PMDB

Valid in the pmd environment

The listpmd command lists information about the PMDB and its subscribers, update file, and error log. If no options are used, the command lists all subscribers of the Policy Model *pmdName.*

This command has the following format:

```
listpmd pmdName \
    [{info|subscriber(subNames)|cmd(offset) \
    |errors|all_errors|log}] \
    [next]
```

cmd*(offset)*

Displays all commands in the update file and their offsets.

The offset indicates the location of the update inside the file. If an offset is specified, the list starts from offset. If no offset is specified, the display begins from the beginning of the update file.

Note: The update file contains updates that must be, or have been, propagated by the PMDB. The offset indicates the location of the next update that must be sent to a subscriber. The update file's initial and latest offsets are displayed.

errors|all_errors

Displays the Policy Model error log. The *errors* parameter displays all types of errors except non-connection failure errors. *all_errors* displays all errors.

info

Displays general information about the Policy Model *pmdName*, including whether the Policy Model has a parent.

next

Display parts of the requested data. This option is useful when the query data is larger than the set query size.

The maximum query size is determined by the query_size configuration setting. The query size default is set at 100.

pmdname

Defines the name of the PMDB you want to list information for.

subscriber(*subNames*)

Lists the subscribers of the Policy Model and their status, including number of errors, availability, offset, and the next command to be propagated. The *subNames* parameter lets you select a subset of subscribers.

log

Displays the policy model general log file.

Example: Display PMDB subscriber information for selected subscribers

To display a list of subscribers to the myPMDB Policy Model that begin with the letters *compInt*, enter the following command:

```
listpmd myPMDB subscriber(compInt*)
```

pmd Command—Control a PMDB

Valid in the pmd environment

The pmd command clears the Policy Model error log, updates the subscriber list, starts and stops the Policy Model service, and truncates the update file.

This command has the following format:

```
pmd pmdName \
    {[release(subname)|start|stop|truncate(offset)|lock|unlock \
    |reloadini|startlog|killog|clrerror|backup|operation]}
```

backup

Moves the Policy Model to backup status.

clrerror|clrerr

Clears the Policy Model error log.

killog

Disables the Policy Model general log file. If you specify this option, no messages are written to the log.

Important! Do not use the kill command to shut down the PMDB service.

lock

Moves the Policy Model to lock status, and stops the Policy Model sending updates to its subscribers.

operation

Moves the Policy Model from backup to operational status.

pmdname

Defines the name of the PMDB that you want to execute the selected option on.

release(*subName*)

Removes the subscriber specified by *subName* from the list of unavailable subscribers. This means that the subscriber can receive updates immediately. *subName* specifies the subscriber that is to become available for update.

reloadini

(UNIX only) Rereads the policy model pmd.ini file and the seos.ini file, letting you change configuration settings without having to reload the policy model daemon.

startlog

Enables the Policy Model general log file for writing. Use this option if the log file has been disabled.

start

Starts the CA Access Control Policy Model service. Use this option when there are no other commands to execute.

stop

Stops the CA Access Control Policy Model daemon/service.

truncate|trunc[(*offset*)]

Deletes entries from the update file. If an offset is not specified, the file is truncated at the highest possible offset. The highest possible offset is the location of last command that successfully updated the subscriber. If *offset* is specified, all the entries up to the specified offset are deleted.

Note: You must now use the true offset provided by the *listpmd* command to truncate the file, and not an offset derived by subtracting from the start offset.

unlock

Moves the Policy Model from lock to unlock status, and lets the Policy Model send updates to its subscribers.

restorepmd Command—Restore a PMDB

Valid in the pmd environment

The restorepmd command restores a PMDB on a local host. The backup files that 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:

restorepmd *pmdName* [source(*path*)] [admin(*user*)] [xadmin(*user*)] [parentpmd(*name*)]

admin(*user*)

(UNIX) Defines internal users as administrators of the restored PMDB.

pmdName

Defines the name of the PMDB to restore.

parentpmd(*name*)

(Optional) Defines the name of the restored PMDB's parent. Specify the name in the format *pmd@host*.

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

xadmin(*user*)

(UNIX) Defines enterprise users as administrators of the restored PMDB.

subs Command—Add Subscribers or Subscribe Databases

Valid in the pmd environment

The subs command adds a subscriber to a parent PMDB or subscribes a database to a parent PMDB.

When you subscribe a host to a PMDB:

- The host must be up.
- CA Access Control must be running on that host
- The PMDB must be the parent PMDB of the subscribed host.

When you subscribe a PMDB to another PMDB:

- The parent_pmd configuration setting of the subscribed PMDB must contain the name of the PMDB to which it is subscribing (its parent PMDB).
- CA Access Control must be running on the host in which the subscribed PMDB resides.

This command has the following format:

subs *pmdname* \
[subs(subsname)] \
[host_type(*mfHost*) sysid(*sysID*) mf_admin(*mfAdmin*) port(*port*)] \
{offset(offset) }

or

subs *pmdname* [newsubs(*subsname*)]

or

subs *pmdname* [parentpmd(*pmdname2@host*)]

host_type(*mfhost*)

The mainframe host type of the subscriber.

mf_admin(*mfAdmin*)

The mainframe administrator of the subscriber.

newsubs(*subsname*)

Subscribes *subname* to policy model *pmdname*, and sends the new subscriber the contents of the whole PMDB, password, and group files.

parentpmd(*pmdName2@host*)

Makes the PMDB *pmdName2@host* the parent Policy Model of *pmdName*.

pmdname

Defines the name of the PMDB you want to execute the selected option on.

port(*port*)

The port number of the subscriber.

subs(*subsname*)

Assigns a subscriber to the PMDB.

sysid(*sysId*)

The system ID of the subscriber.

subspmd Command—Change Parent PMDB

Valid in the pmd environment

The subspmd command changes the parent of the CA Access Control database in the host to which you are connected.

This command has the following format:

subspmd parentpmd(*pmdname@host*)

parentpmd(*pmdname@host*)

Makes *pmdname@host* the current host's parent policy model.

unsubs Command—Remove a Subscriber

Valid in the pmd environment

The unsubs command removes a subscriber from the subscriber list of the Policy Model.

This command has the following format:

unsubs *pmdName* subs(*subName*)

pmdname

Defines the name of the PMDB you want to execute the selected option on.

subs(*subName*)

Defines the name of the subscriber you want to remove from the *pmdname* subscriber list.

Chapter 4: Classes and Properties

This section contains a description of each property in every class defined in the CA Access Control database and in the native operating systems. Arranged in environments alphabetically by class, the chapter provides information on which properties you can modify, which selang parameters you use to update these properties, and which commands contain these parameters.

This section contains the following topics:

Class and Property Information (see page 219)
Classes in the AC Environment (see page 220)
Classes in the Windows Environment (see page 442)
Classes in the UNIX Environment (see page 472)
Classes for Custom Purposes (see page 473)

Class and Property Information

The following conventions apply to the provided class and property information:

- In the descriptive material before the property lists, the *key* of the class record is defined.

  The key is the record identifier, which you specify when you create a new record. Once created, it becomes a non-modifiable property.

- The symbol [-] used with a parameter indicates that the parameter may be deleted from the database by typing it with a minus sign.

  For example, *comment* (with appropriate text) adds a comment to a database record; *comment*- removes the comment from the database. You cannot use parameters with a minus sign when creating a record.

- The two types of classes in the database are accessor classes and resource classes.

  You operate on records in the accessor classes (USER and GROUP) with different selang command sets than you use for the resource classes:

  - Use chusr, editusr, and newusr to operate on USER class records.
  - Use chgrp, editgrp, and newgrp to operate on GROUP class records.
  - Use chres, editres, and newres to operate on records in any of the resource classes. If the resource is a file, you may also use the chfile or editfile commands.
  - Use showgrp, showres, showfile, or showusr to list the properties of a record.
  - Use authorize and authorize- to add, change, or remove ACLs for resource records.

More information:

selang Commands Reference (see page 37)

Classes in the AC Environment

This section contains a complete alphabetic reference to all the classes and properties that exist in the CA Access Control database (classes in the AC environment).

ACVAR Class

Each record in the ACVAR class defines a variable that is deployed on an endpoint. You cannot deactivate this class.

The key of the ACVAR class is the name of the variable.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

OWNER

Defines the user or group that owns the record.

POLICIES

(Informational) The list of policies (POLICY objects) that use this variable.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

VARIABLE_TYPE

Defines the variable type. Valid values are:

built-in

Specifies that the variable was created by CA Access Control during installation. Static variables resolve based on the system settings of the endpoint.

Note: You cannot modify or delete built-in variables.

osvar

Specifies that the variable resolves based on an operating system value.

regval

(Windows) Specifies that the variable resolves based on a registry value.

Note: You can only define registry values that point to REG_SZ or REG_EXPAND_SZ registry types.

static

Specifies that the variable resolves to the string value you define.

Note: You cannot change the variable type of an existing variable.

VARIABLE_VALUE

Defines the values of the variable.

Note: This property does not expand any nested variables within the variable value.

VARIABLE_EXPANDED_VALUE

(Informational) Defines the variable values and expands any nested variables within the variable values.

ADMIN Class

Each record in the ADMIN class contains the definitions that allow non-ADMIN users to administer specific classes. You must create an ADMIN record to represent each CA Access Control class that delegated users will administer. The record contains a list of accessors with the access authorities of each, and also supports conditional access control lists (CACLs).

The key of the ADMIN class record is the name of the class being protected.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked as *informational*.

AAUDIT

(Informational). Displays the type of activity that CA Access Control is auditing.

ACL

Defines a list of accessors (users and groups) permitted to access the resource, and the accessors' access types.

Each element in the access control list (ACL) contains the following information:

Accessor

Defines an accessor.

Access

Defines the access authority that the accessor has to the resource.

Use the access parameter with the authorize or authorize- command to modify the ACL.

CALACL

Defines a list of the accessors (users and groups) that are permitted to access the resource, and their access types according to the Unicenter NSM calendar status.

Each element in the calendar access control list (CALACL) contains the following information:

Accessor

Defines an accessor.

Calendar

Defines a reference to a calendar in Unicenter TNG.

Access

Defines the access authority that the accessor has to the resource.

Access is permitted only when the calendar is ON. Access is denied in all other cases.

Use the calendar parameter with the authorize command to permit user or group access to the resource according to the access defined in the calendar ACL.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

CATEGORY

Defines one or more security categories assigned to a user or a resource.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIME

Defines the day and time restrictions that govern when an accessor can access a resource.

Use the restrictions parameter with the chres, ch[x]usr, or ch[x]grp commands to modify this property.

The resolution of daytime restrictions is one minute.

NACL

The *NACL* property of a resource is an access control list that defines the accessors that are denied authorization to a resource, together with the type of access that they are denied (for example, write). See also ACL, CALACL, PACL. Each entry in the NACL contains the following information:

Accessor

Defines an accessor.

Access

Defines the type of access that is denied to the accessor.

Use the authorize deniedaccess command, or the authorize- deniedaccess- command, to modify this property.

NOTIFY

Defines the user to be notified when a resource or user generates an audit event. CA Access Control can email the audit record to the specified user**.**

Limit: 30 characters.

OWNER

Defines the user or group that owns the record.

PACL

Defines a list of accessors that are permitted to access the resource when the access request is made by a specific program (or a program that matches a name-pattern) and their access types. Each element in the program access control list (PACL) contains the following information:

Accessor

Defines an accessor.

Program

Defines a reference to a record in the PROGRAM class, either specifically or by wildcard pattern matching.

Access

Defines the access authority that the accessor has to the resource.

Note: You can use wildcard characters to specify the resource in a PACL.

Use the via(*pgm*) parameter with the selang authorize command to add programs, accessors, and their access types to a PACL, You can use the authorize- command to remove accessors from a PACL.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

SECLABEL

Defines the security label of a user or resource.

Note: The SECLABEL property corresponds to the label[-] parameter of the chres and ch[x]usr commands.

SECLEVEL

Defines the security level of an accessor or resource.

Note: This property corresponds to the level[-] parameter of the ch[x]usr and chres commands.

UACC

Defines the default access authority for the resource, which indicates the access granted to accessors who are not defined to CA Access Control or who do not appear in the ACL of the resource.

Use the defaccess parameter with the chres, editres, or newres command to modify this property.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

AGENT Class

Each record in the AGENT class defines an object that is used as an agent by CA SSO.

The key of the AGENT class record is the name of the agent.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked as *informational*.

AGENT_TYPE

The type of agent.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

OWNER

Defines the user or group that owns the record.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

AGENT_TYPE Class

Each record in the AGENT_TYPE class defines an agent type used by CA SSO.

The key of the AGENT_TYPE class record is the type of the agent.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked as *informational*.

AGENT_FLAG

Contains information about the attribute. The flag can contain the following values:

- **aznchk-**Indicates whether to use this attribute for authorization.
- **predef** (predefined), **freetext** (free text), or **userdir** (user directory)**-**Specify the source of the user attributes.
- **user** or **group-**These values indicate whether the attribute (accessor) is a user or a group.

AGENT_LIST

A list of objects in the AGENT class that were created with this AGENT_TYPE object as the value for the agent_type parameter; for example, this property is updated implicitly when creating an object in the AGENT class.

CLASSES

A multi-string list of the classes or resources that are relevant to this agent.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

OWNER

Defines the user or group that owns the record.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

APPL Class

Each record in the APPL class defines an application used by CA SSO.

The key of the APPL class record is the name of the application.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked as *informational*.

ACL

Defines a list of accessors (users and groups) permitted to access the resource, and the accessors' access types.

Each element in the access control list (ACL) contains the following information:

Accessor

Defines an accessor.

Access

Defines the access authority that the accessor has to the resource.

Use the access parameter with the authorize or authorize- command to modify the ACL.

APPLTYPE

Used by CA SSO.

AZNACL

Defines the authorization ACL. The authorization ACL is an ACL that allows access to a resource based on the resource description. The description is sent to the authorization engine, not the object. Typically, when an AZNACL is used, the object is not in the database.

CALACL

Defines a list of the accessors (users and groups) that are permitted to access the resource, and their access types according to the Unicenter NSM calendar status.

Each element in the calendar access control list (CALACL) contains the following information:

Accessor

Defines an accessor.

Calendar

Defines a reference to a calendar in Unicenter TNG.

Access

Defines the access authority that the accessor has to the resource.

Access is permitted only when the calendar is ON. Access is denied in all other cases.

Use the calendar parameter with the authorize command to permit user or group access to the resource according to the access defined in the calendar ACL.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

CAPTION

The text under the application's icon on the desktop. The default is the name of the APPL record.

Limit: 47 alphanumeric characters.

CMDLINE

The file name of the application executable. Used by CA SSO.

Limit: 255 characters.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CONTAINED_ITEMS

The record names of the contained applications, if the record is a container.

Use the item[-](*applName*) parameter with the chres, editres, and newres commands to modify this property.

CONTAINERS

The record names of container applications, if the record is contained in other applications.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIME

Defines the day and time restrictions that govern when an accessor can access a resource.

Use the restrictions parameter with the chres, ch[x]usr, or ch[x]grp commands to modify this property.

The resolution of daytime restrictions is one minute.

DIALOG_FILE

The name of the CA SSO script in the directory containing the login sequence for the application. The default directory location is /usr/sso/scripts. The default value is "no script".

Use the script[-](*fileName*) parameter with the chres, editres, and newres commands to modify this property.

GROUPS

A list of user groups authorized to use the application.

HOST

The name of the host where the application resides.

Use the host[-](*hostName*) parameter with the chres, editres, and newres commands to modify this property.

ICONFILE

The file name or full path of the file containing the icon representing the application on the desktop. CA Access Control expects to find the icon on the end user's workstation. If just a file name is entered, the search order for the file is as follows:

1. Current directory
2. Directories listed in the PATH environment variable

The default is the default icon of the workstation.

ICONID

The numeric ID (if necessary) of the icon within the icon file. If the ICONID is not specified, the default icon is used.

IS_CONTAINER

Whether the application is a container. The default is "no".

Use the container[-] parameter with the chres, editres, and newres commands to modify this property.

IS_DISABLED

Whether the application is disabled. If the application is disabled, users cannot log into it. This feature is useful when you change an application and you do not want any users to log in to the application while you make it. The disabled application appears in the application menu list, but if a user selects the application the login is terminated with an appropriate message. The default is "not disabled".

IS_HIDDEN

Whether the application icon appears on the desktop even for users who can invoke it. You may want to hide a *master* application, for example an application that only serves the purpose of supplying passwords to other applications. The default is "not hidden".

Use the hidden[-] parameter with the chres, editres, and newres commands to modify this property.

IS_SENSITIVE

Whether re-authentication is required when the user opens the application after a preset time. The default is "not sensitive".

Use the sensitive[-] parameter with the chres, editres, and newres commands to modify this property.

LOGIN_TYPE

The way user passwords are provided. The value is *pwd* (plain password), *otp* (One Time Password), *appticket* (a proprietary ticket for mainframe application authentication)., *none* (no password required), or *passticket* (a one-time password replacement format created by IBM and used by mainframe security packages). The default is pwd.

Use the login_type(*value*) parameter with the chres, editres, and newres commands to modify this property.

MASTER_APPL

The record name of the application that supplies the password to other applications. The default is no master.

Use the master[-](*applName*) parameter with the chres, editres, and newres commands to modify this property.

NACL

The *NACL* property of a resource is an access control list that defines the accessors that are denied authorization to a resource, together with the type of access that they are denied (for example, write). See also ACL, CALACL, PACL. Each entry in the NACL contains the following information:

Accessor

Defines an accessor.

Access

Defines the type of access that is denied to the accessor.

Use the authorize deniedaccess command, or the authorize- deniedaccess-command, to modify this property.

NOTIFY

Defines the user to be notified when a resource or user generates an audit event. CA Access Control can email the audit record to the specified user**.**

Limit: 30 characters.

OWNER

Defines the user or group that owns the record.

PGMDIR

A directory, or a list of directories, where the application's executable file resides. Used by CA SSO.

PWD_AUTOGEN

Indicates whether the application password is automatically generated by CA SSO. The default is no.

PWD_SYNC

Indicates whether the application password is automatically kept identical to those of the other applications. The default is no.

PWPOLICY

The record name of the password policy for the application. A password policy is a set of rules for checking the validity of a new password and for defining when a password expires. The default is no validity check.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

SCRIPT_POSTCMD

Indicates whether to execute one or more commands after the login script.

SCRIPT_PRECMD

Indicates whether to execute one or more commands before the login script.

SCRIPT_VARS

Used by CA SSO, a variables list with the variable values of the application script that are saved per application.

TKTKEY

Used by CA SSO only.

TKTPROFILE

Used by CA SSO only.

UACC

Defines the default access authority for the resource, which indicates the access granted to accessors who are not defined to CA Access Control or who do not appear in the ACL of the resource.

Use the defaccess parameter with the chres, editres, or newres command to modify this property.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

AUTHHOST Class

Each record in the AUTHHOST class defines an authentication host in CA SSO.

The key of the AUTHHOST class record is the name of the authorization host.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked as *informational*.

ACL

Defines a list of accessors (users and groups) permitted to access the resource, and the accessors' access types.

Each element in the access control list (ACL) contains the following information:

Accessor

Defines an accessor.

Access

Defines the access authority that the accessor has to the resource.

Use the access parameter with the authorize or authorize- command to modify the ACL.

AZNACL

Defines the authorization ACL. The authorization ACL is an ACL that allows access to a resource based on the resource description. The description is sent to the authorization engine, not the object. Typically, when an AZNACL is used, the object is not in the database.

CALACL

Defines a list of the accessors (users and groups) that are permitted to access the resource, and their access types according to the Unicenter NSM calendar status.

Each element in the calendar access control list (CALACL) contains the following information:

Accessor

Defines an accessor.

Calendar

Defines a reference to a calendar in Unicenter TNG.

Access

Defines the access authority that the accessor has to the resource.

Access is permitted only when the calendar is ON. Access is denied in all other cases.

Use the calendar parameter with the authorize command to permit user or group access to the resource according to the access defined in the calendar ACL.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

CATEGORY

Defines one or more security categories assigned to a user or a resource.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIME

Defines the day and time restrictions that govern when an accessor can access a resource.

Use the restrictions parameter with the chres, ch[x]usr, or ch[x]grp commands to modify this property.

The resolution of daytime restrictions is one minute.

ETHINFO

Ethernet information for a host.

GROUPS

The list of GAUTHHOST or CONTAINER records a resource record belongs to.

To modify this property in an AUTHHOST class record, you must change the MEMBERS property in the appropriate CONTAINER or GAUTHHOST record.

Use the mem+ or mem- parameter with the chres, editres or newres command to modify this property.

KEY

Used by CA SSO only.

NACL

The *NACL* property of a resource is an access control list that defines the accessors that are denied authorization to a resource, together with the type of access that they are denied (for example, write). See also ACL, CALACL, PACL. Each entry in the NACL contains the following information:

Accessor

Defines an accessor.

Access

Defines the type of access that is denied to the accessor.

Use the authorize deniedaccess command, or the authorize- deniedaccess- command, to modify this property.

NOTIFY

Defines the user to be notified when a resource or user generates an audit event. CA Access Control can email the audit record to the specified user**.**

Limit: 30 characters.

OWNER

Defines the user or group that owns the record.

PATH

Used by CA SSO only.

PROPERTIES

In UNIX dbdump only

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

SECLABEL

Defines the security label of a user or resource.

Note: The SECLABEL property corresponds to the label[-] parameter of the chres and ch[x]usr commands.

SECLEVEL

Defines the security level of an accessor or resource.

Note: This property corresponds to the level[-] parameter of the ch[x]usr and chres commands.

SEED

Used by CA SSO only.

SERNUM

The serial number of the authentication host.

UACC

Defines the default access authority for the resource, which indicates the access granted to accessors who are not defined to CA Access Control or who do not appear in the ACL of the resource.

Use the defaccess parameter with the chres, editres, or newres command to modify this property.

UNTRUST

Defines whether the resource is untrusted or trusted. If the UNTRUST property is set, accessors cannot use the resource. If the UNTRUST property is not set, the other properties listed in the database for the resource are used to determine accessor's access authority. If a trusted resource is changed in any way, CA Access Control automatically sets the UNTRUST property.

Use the trust[-] parameter with the chres, editres, or newres command to modify this property.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

USER_DIR_PROP

(Informational). The name of the user's directory.

USER_FORMAT

Used by CA SSO only.

USERALIAS

Contains all the user's aliases that are defined to a specific authhost.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

CALENDAR Class

Each record in the CALENDAR class defines a Unicenter TNG calendar object for user, group, and resource enforced time restrictions in CA Access Control. CA Access Control retrieves Unicenter TNG active calendars at specified time intervals for enforcement. Use the calendar(*calendarName*) property of the chgrp, chres, chusr, editgrp, editres, editusr, newgrp, newres, and newusr commands to assign a calendar to a resource.

The following classes have the CALENDAR property in their class records. Each object in any of these resource classes can be assigned *only one* CALENDAR class object.

- ADMIN
- APPL

- AUTHHOST
- CONNECT
- CONTAINER
- DOMAIN (Windows only)
- FILE
- GFILE
- GHOST
- GROUP
- GSUDO
- GTERMINAL
- HOST
- HOSTNET
- HOSTNP
- LOGINAPPL (UNIX only)
- MFTERMINAL
- PROCESS
- PROGRAM
- REGKEY (Windows only)
- SUDO
- SURROGATE
- TCP
- TERMINAL
- USER

The key to the CALENDAR class is the name of the Unicenter TNG calendar.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked as *informational*.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

OWNER

Defines the user or group that owns the record.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

CATEGORY Class

Each record in the CATEGORY class defines a security category in the database.

The key of the CATEGORY class record is the name of the security category.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational.*

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

OWNER

Defines the user or group that owns the record.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

CONNECT Class

Each record in the CONNECT class defines a remote host that can use TCP over IPv4 to connect to that host from the local host.

Note: CA Access Control access rules for IP communication apply only to IPv4. CA Access Control does not control access by IPv6.

Note: If the CONNECT class is being used as a criterion for access, the TCP class cannot effectively control access. Use either the TCP class or the CONECT class to protect a connection, not both.

The key of the CONNECT class record is the name of the remote host.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

ACL

Defines a list of accessors (users and groups) permitted to access the resource, and the accessors' access types.

Each element in the access control list (ACL) contains the following information:

Accessor

Defines an accessor.

Access

Defines the access authority that the accessor has to the resource.

Use the access parameter with the authorize or authorize- command to modify the ACL.

CALACL

Defines a list of the accessors (users and groups) that are permitted to access the resource, and their access types according to the Unicenter NSM calendar status.

Each element in the calendar access control list (CALACL) contains the following information:

Accessor

Defines an accessor.

Calendar

Defines a reference to a calendar in Unicenter TNG.

Access

Defines the access authority that the accessor has to the resource.

Access is permitted only when the calendar is ON. Access is denied in all other cases.

Use the calendar parameter with the authorize command to permit user or group access to the resource according to the access defined in the calendar ACL.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

CATEGORY

Defines one or more security categories assigned to a user or a resource.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIME

Defines the day and time restrictions that govern when an accessor can access a resource.

Use the restrictions parameter with the chres, ch[x]usr, or ch[x]grp commands to modify this property.

The resolution of daytime restrictions is one minute.

GROUPS

Defines the list of CONTAINER records that a resource record belongs to.

To modify this property in a class record, change the MEMBERS property in the appropriate CONTAINER record.

Use the mem+ or mem- parameter with the chres, editres or newres command to modify this property.

NACL

The *NACL* property of a resource is an access control list that defines the accessors that are denied authorization to a resource, together with the type of access that they are denied (for example, write). See also ACL, CALACL, PACL. Each entry in the NACL contains the following information:

Accessor

Defines an accessor.

Access

Defines the type of access that is denied to the accessor.

Use the authorize deniedaccess command, or the authorize- deniedaccess- command, to modify this property.

NOTIFY

Defines the user to be notified when a resource or user generates an audit event. CA Access Control can email the audit record to the specified user**.**

Limit: 30 characters.

OWNER

Defines the user or group that owns the record.

PACL

Defines a list of accessors that are permitted to access the resource when the access request is made by a specific program (or a program that matches a name-pattern) and their access types. Each element in the program access control list (PACL) contains the following information:

Accessor

Defines an accessor.

Program

Defines a reference to a record in the PROGRAM class, either specifically or by wildcard pattern matching.

Access

Defines the access authority that the accessor has to the resource.

Note: You can use wildcard characters to specify the resource in a PACL.

Use the via(*pgm*) parameter with the selang authorize command to add programs, accessors, and their access types to a PACL, You can use the authorize- command to remove accessors from a PACL.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

SECLABEL

Defines the security label of a user or resource.

Note: The SECLABEL property corresponds to the label[-] parameter of the chres and ch[x]usr commands.

SECLEVEL

Defines the security level of an accessor or resource.

Note: This property corresponds to the level[-] parameter of the ch[x]usr and chres commands.

UACC

Defines the default access authority for the resource, which indicates the access granted to accessors who are not defined to CA Access Control or who do not appear in the ACL of the resource.

Use the defaccess parameter with the chres, editres, or newres command to modify this property.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

CONTAINER Class

Each record in the CONTAINER class defines a group of objects from other resource classes, thus simplifying the job of defining access rules when a rule applies to several different classes of objects. Members of a CONTAINER class record can be objects from any of the following classes:

- APPL
- AUTHHOST
- CONNECT
- CONTAINER
- DICTIONARY
- DOMAIN (Windows only)
- FILE
- GAPPL
- GAUTHHOST
- GFILE

- GHOST
- GSUDO
- GTERMINAL
- HNODE
- HOLIDAY
- HOST
- HOSTNET
- HOSTNP
- MFTERMINAL
- PARAM_DESC
- POLICY
- PROCESS
- PROGRAM
- REGKEY (Windows only)
- RULESET
- SPECIALPGM
- SUDO
- SURROGATE
- TCP
- TERMINAL
- WEBSERVICE

Note: CONTAINER records can be nested in other CONTAINER records.

Before you specify an object as a member of a CONTAINER record, you must create a record for it in its appropriate class.

If an object in the container does not have an ACL in its appropriate class record, it inherits the ACL for the CONTAINER record of which it is a member.

The key of the CONTAINER class is the name of the CONTAINER record.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

ACL

Defines a list of accessors (users and groups) permitted to access the resource, and the accessors' access types.

Each element in the access control list (ACL) contains the following information:

Accessor

Defines an accessor.

Access

Defines the access authority that the accessor has to the resource.

Use the access parameter with the authorize or authorize- command to modify the ACL.

CALACL

Defines a list of the accessors (users and groups) that are permitted to access the resource, and their access types according to the Unicenter NSM calendar status.

Each element in the calendar access control list (CALACL) contains the following information:

Accessor

Defines an accessor.

Calendar

Defines a reference to a calendar in Unicenter TNG.

Access

Defines the access authority that the accessor has to the resource.

Access is permitted only when the calendar is ON. Access is denied in all other cases.

Use the calendar parameter with the authorize command to permit user or group access to the resource according to the access defined in the calendar ACL.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIME

Defines the day and time restrictions that govern when an accessor can access a resource.

Use the restrictions parameter with the chres, ch[x]usr, or ch[x]grp commands to modify this property.

The resolution of daytime restrictions is one minute.

GROUPS

Defines the list of CONTAINER records that a resource record belongs to.

To modify this property in a class record, change the MEMBERS property in the appropriate CONTAINER record.

Use the mem+ or mem- parameter with the chres, editres or newres command to modify this property.

MEMBERS

The list of objects from any class that are members of the group.

Use the mem+ or mem- parameter with the chres, editres, and newres commands to modify this property.

NACL

The *NACL* property of a resource is an access control list that defines the accessors that are denied authorization to a resource, together with the type of access that they are denied (for example, write). See also ACL, CALACL, PACL. Each entry in the NACL contains the following information:

Accessor

Defines an accessor.

Access

Defines the type of access that is denied to the accessor.

Use the authorize deniedaccess command, or the authorize- deniedaccess- command, to modify this property.

OWNER

Defines the user or group that owns the record.

PACL

Defines a list of accessors that are permitted to access the resource when the access request is made by a specific program (or a program that matches a name-pattern) and their access types. Each element in the program access control list (PACL) contains the following information:

Accessor

Defines an accessor.

Program

Defines a reference to a record in the PROGRAM class, either specifically or by wildcard pattern matching.

Access

Defines the access authority that the accessor has to the resource.

Note: You can use wildcard characters to specify the resource in a PACL.

Use the via(*pgm*) parameter with the selang authorize command to add programs, accessors, and their access types to a PACL, You can use the authorize- command to remove accessors from a PACL.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

DEPLOYMENT Class

Each record in the DEPLOYMENT class defines a deployment or undeployment task for an endpoint. A deployment task includes information that the endpoint needs to deploy or undeploy a policy as required.

The key of the DEPLOYMENT class is the name of the deployment task, which is usually generated automatically.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

ACL

Defines a list of accessors (users and groups) permitted to access the resource, and the accessors' access types.

Each element in the access control list (ACL) contains the following information:

Accessor

Defines an accessor.

Access

Defines the access authority that the accessor has to the resource.

Use the access parameter with the authorize or authorize- command to modify the ACL.

CALACL

Defines a list of the accessors (users and groups) that are permitted to access the resource, and their access types according to the Unicenter NSM calendar status.

Each element in the calendar access control list (CALACL) contains the following information:

Accessor

Defines an accessor.

Calendar

Defines a reference to a calendar in Unicenter TNG.

Access

Defines the access authority that the accessor has to the resource.

Access is permitted only when the calendar is ON. Access is denied in all other cases.

Use the calendar parameter with the authorize command to permit user or group access to the resource according to the access defined in the calendar ACL.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

CATEGORY

Defines one or more security categories assigned to a user or a resource.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIME

Defines the day and time restrictions that govern when an accessor can access a resource.

Use the restrictions parameter with the chres, ch[x]usr, or ch[x]grp commands to modify this property.

The resolution of daytime restrictions is one minute.

DMS_NAME

Defines the name of the DMS where the deployment task was created.

GPOLICY

Defines the name of the policy this deployment task was created for.

GROUPS

Defines the deployment package (GDEPLOYMENT) this deployment task is a member of.

HNODE

Defines the host this deployment task was created for.

NACL

The *NACL* property of a resource is an access control list that defines the accessors that are denied authorization to a resource, together with the type of access that they are denied (for example, write). See also ACL, CALACL, PACL. Each entry in the NACL contains the following information:

Accessor

Defines an accessor.

Access

Defines the type of access that is denied to the accessor.

Use the authorize deniedaccess command, or the authorize- deniedaccess-command, to modify this property.

NOTIFY

Defines the user to be notified when a resource or user generates an audit event. CA Access Control can email the audit record to the specified user**.**

Limit: 30 characters.

OPERATION

Specifies the type of operation that the endpoint should perform as a result of this deployment task. Can be one of: Deploy, Undeploy.

OWNER

Defines the user or group that owns the record.

PACL

Defines a list of accessors that are permitted to access the resource when the access request is made by a specific program (or a program that matches a name-pattern) and their access types. Each element in the program access control list (PACL) contains the following information:

Accessor

Defines an accessor.

Program

Defines a reference to a record in the PROGRAM class, either specifically or by wildcard pattern matching.

Access

Defines the access authority that the accessor has to the resource.

Note: You can use wildcard characters to specify the resource in a PACL.

Use the via(*pgm*) parameter with the selang authorize command to add programs, accessors, and their access types to a PACL, You can use the authorize- command to remove accessors from a PACL.

POLICY_VERSION

Defines the policy version this deployment task was created for.

RESULT_MESSAGE

Defines the output from the deployment or undeployment selang script. These are the messages selang outputs when the policy deployment or undeployment scripts are run.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

SECLABEL

Defines the security label of a user or resource.

Note: The SECLABEL property corresponds to the label[-] parameter of the chres and ch[x]usr commands.

SECLEVEL

Defines the security level of an accessor or resource.

Note: This property corresponds to the level[-] parameter of the ch[x]usr and chres commands.

STATUS

Defines the status of the deployment task. Can be one of:

- **Success**—Policy deployed without errors.
- **Warning**—Deployment script executed with errors.
- **Fail**—There was an error executing the deployment task.
- **No Action**—The deployment package is effectively empty so there is nothing to do.

  Note: This can also be a result of the policy already being assigned to the host through a different deployment path.
- **Not Executed**—Policy verification found one or more errors in the policy.
- **Out of Sync**—The policy contains a variable and the variable value has changed on the endpoint.
- **Pending Deployment**—The policy contains an undefined or unresolved variable.
- **Pending Prerequisite**—The deployment task will be executed only after all prerequisite policies are deployed.
- **Pending Dependents**—The deployment task will be executed (undeploy policy) only after all dependent policies are also undeployed.
- **Fix**—The deployment task is waiting to be deployed again.

TARGETYPE

Defines the type of host (target) to limit the policyfetcher to process only CA Access Control deployment packages. Can be one of: UNAB, AC, None.

UACC

Defines the default access authority for the resource, which indicates the access granted to accessors who are not defined to CA Access Control or who do not appear in the ACL of the resource.

Use the defaccess parameter with the chres, editres, or newres command to modify this property.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

DICTIONARY Class

Each record in the DICTIONARY class defines a word in a common dictionary stored in the CA Access Control database to compare passwords to. When users change their passwords, the passwords are checked against each record in this DICTIONARY class.

In addition to adding records (words) to the DICTIONARY class, you can import dictionary words from external files by running a utility or program.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

OWNER

Defines the user or group that owns the record.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

DOMAIN Class

Valid on Windows

Each record in the DOMAIN class defines a domain in the Windows network.

The key to the DOMAIN record is the domain name.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

ACL

Defines a list of accessors (users and groups) permitted to access the resource, and the accessors' access types.

Each element in the access control list (ACL) contains the following information:

Accessor

Defines an accessor.

Access

Defines the access authority that the accessor has to the resource.

Use the access parameter with the authorize or authorize- command to modify the ACL.

CALACL

Defines a list of the accessors (users and groups) that are permitted to access the resource, and their access types according to the Unicenter NSM calendar status.

Each element in the calendar access control list (CALACL) contains the following information:

Accessor

Defines an accessor.

Calendar

Defines a reference to a calendar in Unicenter TNG.

Access

Defines the access authority that the accessor has to the resource.

Access is permitted only when the calendar is ON. Access is denied in all other cases.

Use the calendar parameter with the authorize command to permit user or group access to the resource according to the access defined in the calendar ACL.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

CATEGORY

Defines one or more security categories assigned to a user or a resource.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIME

Defines the day and time restrictions that govern when an accessor can access a resource.

Use the restrictions parameter with the chres, ch[x]usr, or ch[x]grp commands to modify this property.

The resolution of daytime restrictions is one minute.

GROUPS

Defines the list of CONTAINER records that a resource record belongs to.

To modify this property in a class record, change the MEMBERS property in the appropriate CONTAINER record.

Use the mem+ or mem- parameter with the chres, editres or newres command to modify this property.

NACL

The *NACL* property of a resource is an access control list that defines the accessors that are denied authorization to a resource, together with the type of access that they are denied (for example, write). See also ACL, CALACL, PACL. Each entry in the NACL contains the following information:

Accessor

Defines an accessor.

Access

Defines the type of access that is denied to the accessor.

Use the authorize deniedaccess command, or the authorize- deniedaccess- command, to modify this property.

NOTIFY

Defines the user to be notified when a resource or user generates an audit event. CA Access Control can email the audit record to the specified user**.**

Limit: 30 characters.

OWNER

Defines the user or group that owns the record.

PACL

Defines a list of accessors that are permitted to access the resource when the access request is made by a specific program (or a program that matches a name-pattern) and their access types. Each element in the program access control list (PACL) contains the following information:

Accessor

Defines an accessor.

Program

Defines a reference to a record in the PROGRAM class, either specifically or by wildcard pattern matching.

Access

Defines the access authority that the accessor has to the resource.

Note: You can use wildcard characters to specify the resource in a PACL.

Use the via(*pgm*) parameter with the selang authorize command to add programs, accessors, and their access types to a PACL, You can use the authorize- command to remove accessors from a PACL.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

SECLABEL

Defines the security label of a user or resource.

Note: The SECLABEL property corresponds to the label[-] parameter of the chres and ch[x]usr commands.

SECLEVEL

Defines the security level of an accessor or resource.

Note: This property corresponds to the level[-] parameter of the ch[x]usr and chres commands.

UACC

Defines the default access authority for the resource, which indicates the access granted to accessors who are not defined to CA Access Control or who do not appear in the ACL of the resource.

Use the defaccess parameter with the chres, editres, or newres command to modify this property.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

FILE Class

Each record in the FILE class defines the access allowed to a specific file or directory, or to files that match a *file name pattern*. A file need not have been created yet in order to have a rule defined for it.

Device files and symbolic links can be protected like any other file. However, by protecting a link you do not automatically protect the file that the link points to.

Note: On the NTFS file system, a record in the FILE class also defines the access to the file's streams. For more information about how you can protect file streams, see the *Endpoint Administration Guide for Windows*.

When you define a script as a file, allow both *read* and *execute* access to the file. When you define a binary, *execute* access is sufficient.

For users outside the special _restricted group, the _default record in the FILE class (or if no _default record exists, the record for FILE in the UACC class) *protects only files that are part of CA Access Control*-such as the seos.ini, seosd.trace, seos.audit, and seos.error files. These files are not explicitly defined to CA Access Control, but are automatically protected by CA Access Control.

The key of the FILE class record is the name of the file or directory protected by the record. The full path must be specified.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

ACL

Defines a list of accessors (users and groups) permitted to access the resource, and the accessors' access types.

Each element in the access control list (ACL) contains the following information:

Accessor

Defines an accessor.

Access

Defines the access authority that the accessor has to the resource.

Use the access parameter with the authorize or authorize- command to modify the ACL.

CALACL

Defines a list of the accessors (users and groups) that are permitted to access the resource, and their access types according to the Unicenter NSM calendar status.

Each element in the calendar access control list (CALACL) contains the following information:

Accessor

Defines an accessor.

Calendar

Defines a reference to a calendar in Unicenter TNG.

Access

Defines the access authority that the accessor has to the resource.

Access is permitted only when the calendar is ON. Access is denied in all other cases.

Use the calendar parameter with the authorize command to permit user or group access to the resource according to the access defined in the calendar ACL.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

CATEGORY

Defines one or more security categories assigned to a user or a resource.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIME

Defines the day and time restrictions that govern when an accessor can access a resource.

Use the restrictions parameter with the chres, ch[x]usr, or ch[x]grp commands to modify this property.

The resolution of daytime restrictions is one minute.

Groups

The list of GFILE or CONTAINER records a resource record belongs to.

DB property: GROUPS

To modify this property in a FILE class record, you must change the MEMBERS property in the appropriate CONTAINER or GFILE record.

Use the mem+ or mem- parameter with the chres, editres or newres command to modify this property.

NACL

The *NACL* property of a resource is an access control list that defines the accessors that are denied authorization to a resource, together with the type of access that they are denied (for example, write). See also ACL, CALACL, PACL. Each entry in the NACL contains the following information:

Accessor

Defines an accessor.

Access

Defines the type of access that is denied to the accessor.

Use the authorize deniedaccess command, or the authorize- deniedaccess- command, to modify this property.

NOTIFY

Defines the user to be notified when a resource or user generates an audit event. CA Access Control can email the audit record to the specified user**.**

Limit: 30 characters.

OWNER

Defines the user or group that owns the record.

PACL

Defines a list of accessors that are permitted to access the resource when the access request is made by a specific program (or a program that matches a name-pattern) and their access types. Each element in the program access control list (PACL) contains the following information:

Accessor

Defines an accessor.

Program

Defines a reference to a record in the PROGRAM class, either specifically or by wildcard pattern matching.

Access

Defines the access authority that the accessor has to the resource.

Note: You can use wildcard characters to specify the resource in a PACL.

Use the via(*pgm*) parameter with the selang authorize command to add programs, accessors, and their access types to a PACL, You can use the authorize- command to remove accessors from a PACL.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

SECLABEL

Defines the security label of a user or resource.

Note: The SECLABEL property corresponds to the label[-] parameter of the chres and ch[x]usr commands.

SECLEVEL

Defines the security level of an accessor or resource.

Note: This property corresponds to the level[-] parameter of the ch[x]usr and chres commands.

UACC

Defines the default access authority for the resource, which indicates the access granted to accessors who are not defined to CA Access Control or who do not appear in the ACL of the resource.

Use the defaccess parameter with the chres, editres, or newres command to modify this property.

UNTRUST

Defines whether the resource is untrusted or trusted. If the UNTRUST property is set, accessors cannot use the resource. If the UNTRUST property is not set, the other properties listed in the database for the resource are used to determine accessor's access authority. If a trusted resource is changed in any way, CA Access Control automatically sets the UNTRUST property.

Use the trust[-] parameter with the chres, editres, or newres command to modify this property.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

GAPPL Class

Each record in the GAPPL class defines a group of applications used by CA SSO. You must create an APPL class record for each application before adding it to a GAPPL record. You must then explicitly connect records of the APPL class to the GAPPL record in order to group them.

The key of the GAPPL class record is the name of the GAPPL record.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces.

ACL

Defines a list of accessors (users and groups) permitted to access the resource, and the accessors' access types.

Each element in the access control list (ACL) contains the following information:

Accessor

Defines an accessor.

Access

Defines the access authority that the accessor has to the resource.

Use the access parameter with the authorize or authorize- command to modify the ACL.

AZNACL

Defines the authorization ACL. The authorization ACL is an ACL that allows access to a resource based on the resource description. The description is sent to the authorization engine, not the object. Typically, when an AZNACL is used, the object is not in the database.

CALACL

Defines a list of the accessors (users and groups) that are permitted to access the resource, and their access types according to the Unicenter NSM calendar status.

Each element in the calendar access control list (CALACL) contains the following information:

Accessor

Defines an accessor.

Calendar

Defines a reference to a calendar in Unicenter TNG.

Access

Defines the access authority that the accessor has to the resource.

Access is permitted only when the calendar is ON. Access is denied in all other cases.

Use the calendar parameter with the authorize command to permit user or group access to the resource according to the access defined in the calendar ACL.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

GROUPS

The list of CONTAINER records a resource record belongs to.

To modify this property in a GAPPL class record, you must change the MEMBERS property in the appropriate CONTAINER record.

Use the mem+ or mem- parameter with the chres, editres or newres command to modify this property.

MEMBERS

The list of objects from the APPL class that are members of the group.

Use the mem+ or mem- parameter with the chres, editres, and newres commands to modify this property.

NACL

The *NACL* property of a resource is an access control list that defines the accessors that are denied authorization to a resource, together with the type of access that they are denied (for example, write). See also ACL, CALACL, PACL. Each entry in the NACL contains the following information:

Accessor

Defines an accessor.

Access

Defines the type of access that is denied to the accessor.

Use the authorize deniedaccess command, or the authorize- deniedaccess-command, to modify this property.

OWNER

Defines the user or group that owns the record.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

GAUTHHOST Class

Each record in the GAUTHHOST class defines a group of authentication hosts used by CA SSO. You must create an AUTHHOST class record for each application before adding it to a GAUTHHOST record. You must then explicitly connect records of the AUTHHOST class to the GAUTHHOST record in order to group them.

The key of the GAUTHHOST class record is the name of the GAUTHHOST record.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

ACL

Defines a list of accessors (users and groups) permitted to access the resource, and the accessors' access types.

Each element in the access control list (ACL) contains the following information:

Accessor

Defines an accessor.

Access

Defines the access authority that the accessor has to the resource.

Use the access parameter with the authorize or authorize- command to modify the ACL.

AZNACL

Defines the authorization ACL. The authorization ACL is an ACL that allows access to a resource based on the resource description. The description is sent to the authorization engine, not the object. Typically, when an AZNACL is used, the object is not in the database.

CALACL

Defines a list of the accessors (users and groups) that are permitted to access the resource, and their access types according to the Unicenter NSM calendar status.

Each element in the calendar access control list (CALACL) contains the following information:

Accessor

Defines an accessor.

Calendar

Defines a reference to a calendar in Unicenter TNG.

Access

Defines the access authority that the accessor has to the resource.

Access is permitted only when the calendar is ON. Access is denied in all other cases.

Use the calendar parameter with the authorize command to permit user or group access to the resource according to the access defined in the calendar ACL.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

GROUPS

The list of CONTAINER records a resource record belongs to.

To modify this property in a GAUTHHOST class record, you must change the MEMBERS property in the appropriate CONTAINER record.

Use the mem+ or mem- parameter with the chres, editres or newres command to modify this property.

MEMBERS

The list of objects from the AUTHHOST class that are members of the group.

Use the mem+ or mem- parameter with the chres, editres, and newres commands to modify this property.

NACL

The *NACL* property of a resource is an access control list that defines the accessors that are denied authorization to a resource, together with the type of access that they are denied (for example, write). See also ACL, CALACL, PACL. Each entry in the NACL contains the following information:

Accessor

Defines an accessor.

Access

Defines the type of access that is denied to the accessor.

Use the authorize deniedaccess command, or the authorize- deniedaccess-command, to modify this property.

OWNER

Defines the user or group that owns the record.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

GFILE Class

Each record in the GFILE class defines the access allowed to a group of specific files, specific directories, or files that match a name pattern. You must create a FILE class record for each application before adding it to a GFILE record. You must then explicitly connect records of the FILE class to the GFILE record in order to group them. A file need not have been created yet in order to have a FILE class record defined for it.

The key of the GFILE class record is the name of the GFILE record.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

ACL

Defines a list of accessors (users and groups) permitted to access the resource, and the accessors' access types.

Each element in the access control list (ACL) contains the following information:

Accessor

Defines an accessor.

Access

Defines the access authority that the accessor has to the resource.

Use the access parameter with the authorize or authorize- command to modify the ACL.

CALACL

Defines a list of the accessors (users and groups) that are permitted to access the resource, and their access types according to the Unicenter NSM calendar status.

Each element in the calendar access control list (CALACL) contains the following information:

Accessor

Defines an accessor.

Calendar

Defines a reference to a calendar in Unicenter TNG.

Access

Defines the access authority that the accessor has to the resource.

Access is permitted only when the calendar is ON. Access is denied in all other cases.

Use the calendar parameter with the authorize command to permit user or group access to the resource according to the access defined in the calendar ACL.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIME

Defines the day and time restrictions that govern when an accessor can access a resource.

Use the restrictions parameter with the chres, ch[x]usr, or ch[x]grp commands to modify this property.

The resolution of daytime restrictions is one minute.

GROUPS

Defines the list of CONTAINER records that a resource record belongs to.

To modify this property in a class record, change the MEMBERS property in the appropriate CONTAINER record.

Use the mem+ or mem- parameter with the chres, editres or newres command to modify this property.

MEMBERS

The list of objects from the FILE class that are members of the group.

Use the mem+ or mem- parameter with the chres, editres, and newres commands to modify this property.

NACL

The *NACL* property of a resource is an access control list that defines the accessors that are denied authorization to a resource, together with the type of access that they are denied (for example, write). See also ACL, CALACL, PACL. Each entry in the NACL contains the following information:

Accessor

Defines an accessor.

Access

Defines the type of access that is denied to the accessor.

Use the authorize deniedaccess command, or the authorize- deniedaccess-command, to modify this property.

NOTIFY

Defines the user to be notified when a resource or user generates an audit event. CA Access Control can email the audit record to the specified user**.**

Limit: 30 characters.

OWNER

Defines the user or group that owns the record.

PACL

Defines a list of accessors that are permitted to access the resource when the access request is made by a specific program (or a program that matches a name-pattern) and their access types. Each element in the program access control list (PACL) contains the following information:

Accessor

Defines an accessor.

Program

Defines a reference to a record in the PROGRAM class, either specifically or by wildcard pattern matching.

Access

Defines the access authority that the accessor has to the resource.

Note: You can use wildcard characters to specify the resource in a PACL.

Use the via(*pgm*) parameter with the selang authorize command to add programs, accessors, and their access types to a PACL, You can use the authorize- command to remove accessors from a PACL.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

GDEPLOYMENT Class

Each record in the GDEPLOYMENT class defines a deployment package. A deployment package is created automatically on the DMS and groups together all the deployment tasks that are created as a result of the same transaction (policy assignment, upgrade, and so on) and for a particular host. This means that each transaction you make creates the required number of deployment tasks (DEPLOYMENT objects) and groups these by host (GDEPLOYMENT objects).

The key of the GDEPLOYMENT class is the name of the deployment package, which is usually generated automatically.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

ACL

Defines a list of accessors (users and groups) permitted to access the resource, and the accessors' access types.

Each element in the access control list (ACL) contains the following information:

Accessor

Defines an accessor.

Access

Defines the access authority that the accessor has to the resource.

Use the access parameter with the authorize or authorize- command to modify the ACL.

CALACL

Defines a list of the accessors (users and groups) that are permitted to access the resource, and their access types according to the Unicenter NSM calendar status.

Each element in the calendar access control list (CALACL) contains the following information:

Accessor

Defines an accessor.

Calendar

Defines a reference to a calendar in Unicenter TNG.

Access

Defines the access authority that the accessor has to the resource.

Access is permitted only when the calendar is ON. Access is denied in all other cases.

Use the calendar parameter with the authorize command to permit user or group access to the resource according to the access defined in the calendar ACL.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

CATEGORY

Defines one or more security categories assigned to a user or a resource.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIME

Defines the day and time restrictions that govern when an accessor can access a resource.

Use the restrictions parameter with the chres, ch[x]usr, or ch[x]grp commands to modify this property.

The resolution of daytime restrictions is one minute.

GHNODE

Defines the host group this deployment package was created for.

GROUPS

Defines the list of CONTAINER records that a resource record belongs to.

To modify this property in a class record, change the MEMBERS property in the appropriate CONTAINER record.

Use the mem+ or mem- parameter with the chres, editres or newres command to modify this property.

HNODE

Defines the host this deployment package was created for.

MEMBERS

The list of objects from the DEPLOYMENT class that are members of the group.

Use the mem+ or mem- parameter with the chres, editres, and newres commands to modify this property.

NACL

The *NACL* property of a resource is an access control list that defines the accessors that are denied authorization to a resource, together with the type of access that they are denied (for example, write). See also ACL, CALACL, PACL. Each entry in the NACL contains the following information:

Accessor

Defines an accessor.

Access

Defines the type of access that is denied to the accessor.

Use the authorize deniedaccess command, or the authorize- deniedaccess- command, to modify this property.

NOTIFY

Defines the user to be notified when a resource or user generates an audit event. CA Access Control can email the audit record to the specified user**.**

Limit: 30 characters.

OWNER

Defines the user or group that owns the record.

PACL

Defines a list of accessors that are permitted to access the resource when the access request is made by a specific program (or a program that matches a name-pattern) and their access types. Each element in the program access control list (PACL) contains the following information:

Accessor

Defines an accessor.

Program

Defines a reference to a record in the PROGRAM class, either specifically or by wildcard pattern matching.

Access

Defines the access authority that the accessor has to the resource.

Note: You can use wildcard characters to specify the resource in a PACL.

Use the via(*pgm*) parameter with the selang authorize command to add programs, accessors, and their access types to a PACL, You can use the authorize- command to remove accessors from a PACL.

POLICY

Defines the policy this deployment package was created for.

SECLABEL

Defines the security label of a user or resource.

Note: The SECLABEL property corresponds to the label[-] parameter of the chres and ch[x]usr commands.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

SECLEVEL

Defines the security level of an accessor or resource.

Note: This property corresponds to the level[-] parameter of the ch[x]usr and chres commands.

TRIGGER

Specifies the reason this deployment package was created. Can be one of:

- Assign—A result of assigning a policy to a host, or a host to a host group.
- UnAssign—A result of unassigning a policy from a host, or a host from a host group.
- Direct Deploy—A result of a direct deploy action.
- Direct Undeploy—A result of a direct undeploy action.
- Upgrade—A result of an upgrade action.
- Restore—A result of a restore action on a host (HNODE).
- Hnode Deletion—-A result of deleting a host (HNODE).
- Ghnode Deletion—A result of deleting a host group (GHNODE).
- Reset—A result of resetting a host.
- Downgrade—A result of policy downgrade on hosts.

UACC

Defines the default access authority for the resource, which indicates the access granted to accessors who are not defined to CA Access Control or who do not appear in the ACL of the resource.

Use the defaccess parameter with the chres, editres, or newres command to modify this property.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

GHNODE Class

Each record in the GHNODE class defines a host group-a group of hosts (HNODE objects). You must create a HNODE class record for each host before adding it to a GHOST record.

This class is used to manage policy deployment and assignment.

The key of the GHNODE class record is a logical name for the host group.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

ACL

Defines a list of accessors (users and groups) permitted to access the resource, and the accessors' access types.

Each element in the access control list (ACL) contains the following information:

Accessor

Defines an accessor.

Access

Defines the access authority that the accessor has to the resource.

Use the access parameter with the authorize or authorize- command to modify the ACL.

CALACL

Defines a list of the accessors (users and groups) that are permitted to access the resource, and their access types according to the Unicenter NSM calendar status.

Each element in the calendar access control list (CALACL) contains the following information:

Accessor

Defines an accessor.

Calendar

Defines a reference to a calendar in Unicenter TNG.

Access

Defines the access authority that the accessor has to the resource.

Access is permitted only when the calendar is ON. Access is denied in all other cases.

Use the calendar parameter with the authorize command to permit user or group access to the resource according to the access defined in the calendar ACL.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

GROUPS

Defines the list of CONTAINER records that a resource record belongs to.

To modify this property in a class record, change the MEMBERS property in the appropriate CONTAINER record.

Use the mem+ or mem- parameter with the chres, editres or newres command to modify this property.

MEMBERS

The list of objects from the HNODE class that are members of the group.

Use the mem+ or mem- parameter with the chres, editres, and newres commands to modify this property.

NACL

The *NACL* property of a resource is an access control list that defines the accessors that are denied authorization to a resource, together with the type of access that they are denied (for example, write). See also ACL, CALACL, PACL. Each entry in the NACL contains the following information:

Accessor

Defines an accessor.

Access

Defines the type of access that is denied to the accessor.

Use the authorize deniedaccess command, or the authorize- deniedaccess-command, to modify this property.

OWNER

Defines the user or group that owns the record.

POLICIES

The list of policies that should be deployed on this object.

POLICYASSIGN

Defines the list of policies that are assigned to this object.

Display name: Assigned Policies

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

UACC

Defines the default access authority for the resource, which indicates the access granted to accessors who are not defined to CA Access Control or who do not appear in the ACL of the resource.

Use the defaccess parameter with the chres, editres, or newres command to modify this property.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

GHOST Class

Each record in the GHOST class defines a group of hosts. You must create a HOST class record for each host before adding it to a GHOST record. The services must be defined to the system using the /etc/services file (for UNIX), \system32\drivers\etc\services file (for Windows), or another service name resolution method. When authorizing services, you may identify the services by their port numbers in the TCP/IP protocol rather than by their names. When adding services, you may identify the services by their port numbers in the TCP/IP protocol rather than by their names. You must then explicitly connect records of the HOST class to the GHOST record in order to group them.

GHOST records define access rules that govern the access other stations (hosts) belonging to the group of hosts have to the local host when using Internet communication. For each client group (GHOST record), the INETACL property lists the service rules that govern the services the local host may provide to hosts belonging to the client group.

The key of the GHOST class record is the name of the GHOST record.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIME

Defines the day and time restrictions that govern when an accessor can access a resource.

Use the restrictions parameter with the chres, ch[x]usr, or ch[x]grp commands to modify this property.

The resolution of daytime restrictions is one minute.

GROUPS

Defines the list of CONTAINER records that a resource record belongs to.

To modify this property in a class record, change the MEMBERS property in the appropriate CONTAINER record.

Use the mem+ or mem- parameter with the chres, editres or newres command to modify this property.

INETACL

Defines the services the local host is allowed to provide to the group of client hosts and what their access types are. Each element in the access control list contains the following information:

Services reference

A reference to a service (a port number or name). To specify all the services, enter an asterisk (*) as the services reference.

CA Access Control supports dynamic port names as specified in the /etc/rpc file (for UNIX) or \etc\rpc file (for Windows).

Access

Defines the access authority that the accessor has to the resource.

Use the access(*type-of-access*), service, and stationName parameters with the authorize[-] command to modify accessors and their access types in the INETACL property.

INSERVRNGE

Specifies the range of services that the local host provides to the group of client hosts.

Performs a similar function to the INETACL property.

Use the service(*serviceRange*) parameter with the authorize[-] command to modify accessors and their access types in the INSERVRANGE property.

MEMBERS

The list of objects from the HOST class that are members of the group.

Use the mem+ or mem- parameter with the chres, editres, and newres commands to modify this property.

OWNER

Defines the user or group that owns the record.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

GPOLICY Class

Each record in the GPOLICY class defines a logical policy. It contains information about the policy versions (POLICY objects) that belong to this policy and the hosts and host groups it is assigned to.

The key of the GDEPLOYMENT class is the name of the logical policy.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

ACL

Defines a list of accessors (users and groups) permitted to access the resource, and the accessors' access types.

Each element in the access control list (ACL) contains the following information:

Accessor

Defines an accessor.

Access

Defines the access authority that the accessor has to the resource.

Use the access parameter with the authorize or authorize- command to modify the ACL.

CALACL

Defines a list of the accessors (users and groups) that are permitted to access the resource, and their access types according to the Unicenter NSM calendar status.

Each element in the calendar access control list (CALACL) contains the following information:

Accessor

Defines an accessor.

Calendar

Defines a reference to a calendar in Unicenter TNG.

Access

Defines the access authority that the accessor has to the resource.

Access is permitted only when the calendar is ON. Access is denied in all other cases.

Use the calendar parameter with the authorize command to permit user or group access to the resource according to the access defined in the calendar ACL.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

CATEGORY

Defines one or more security categories assigned to a user or a resource.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIME

Defines the day and time restrictions that govern when an accessor can access a resource.

Use the restrictions parameter with the chres, ch[x]usr, or ch[x]grp commands to modify this property.

The resolution of daytime restrictions is one minute.

GHNODEASSIGN

Defines the host groups this policy is assigned to.

GROUPS

Defines the list of CONTAINER records that a resource record belongs to.

To modify this property in a class record, change the MEMBERS property in the appropriate CONTAINER record.

Use the mem+ or mem- parameter with the chres, editres or newres command to modify this property.

HNODEASSIGN

Defines the hosts this policy is assigned to.

LATEST_FINALIZED_VERSION

Defines the name of the latest policy version (POLICY object) that is finalized.

LATEST_VERSION

Defines the name of the latest policy version (POLICY object) associated with this policy.

MEMBERS

The list of objects from the POLICY class (policy versions) that are members of the group.

Use the mem+ or mem- parameter with the chres, editres, and newres commands to modify this property.

NACL

The *NACL* property of a resource is an access control list that defines the accessors that are denied authorization to a resource, together with the type of access that they are denied (for example, write). See also ACL, CALACL, PACL. Each entry in the NACL contains the following information:

Accessor

Defines an accessor.

Access

Defines the type of access that is denied to the accessor.

Use the authorize deniedaccess command, or the authorize- deniedaccess- command, to modify this property.

NOTIFY

Defines the user to be notified when a resource or user generates an audit event. CA Access Control can email the audit record to the specified user**.**

Limit: 30 characters.

OWNER

Defines the user or group that owns the record.

PACL

Defines a list of accessors that are permitted to access the resource when the access request is made by a specific program (or a program that matches a name-pattern) and their access types. Each element in the program access control list (PACL) contains the following information:

Accessor

Defines an accessor.

Program

Defines a reference to a record in the PROGRAM class, either specifically or by wildcard pattern matching.

Access

Defines the access authority that the accessor has to the resource.

Note: You can use wildcard characters to specify the resource in a PACL.

Use the via(*pgm*) parameter with the selang authorize command to add programs, accessors, and their access types to a PACL, You can use the authorize- command to remove accessors from a PACL.

POLICY TYPE

A value representing the group policy type. Valid values are:

- None
- Login—Specifies that the policy is a UNAB login policy.
- Configuration—Specifies that the policy is a UNAB configuration policy.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

SECLABEL

Defines the security label of a user or resource.

Note: The SECLABEL property corresponds to the label[-] parameter of the chres and ch[x]usr commands.

SECLEVEL

Defines the security level of an accessor or resource.

Note: This property corresponds to the level[-] parameter of the ch[x]usr and chres commands.

UACC

Defines the default access authority for the resource, which indicates the access granted to accessors who are not defined to CA Access Control or who do not appear in the ACL of the resource.

Use the defaccess parameter with the chres, editres, or newres command to modify this property.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

GROUP Class

Each record in the GROUP class defines a group of users in the database.

The key of each GROUP class record is the name of the group.

Note: The properties of profile groups apply to each user associated with the profile group. However, if the same property is specified in a user (USER or XUSER) record, the user record overrides those in the profile group record.

You can change most of these properties from the CA Access Control Endpoint Management, or by using the selang command chgrp.

Note: In most cases, and unless otherwise indicated, to change a property using ch[x]grp, you use the property name as the command parameter.

You can view all properties from the CA Access Control Endpoint Management, or by using the selang command showgrp.

APPLS

(Informational) Displays the list of applications that the accessor is authorized to access. Used by CA SSO.

AUDIT_MODE

Defines the activities that CA Access Control records in the audit log. You can specify any combination of the following activities:

- No logging
- All activities recorded in the trace file
- Unsuccessful login attempts
- Successful logins
- Failed access attempts to resources protected by CA Access Control
- Successful accesses to resources protected by CA Access Control

Note: This property corresponds to the audit parameter of the ch[x]usr and ch[x]grp commands. You can use AUDIT_MODE for a GROUP or XGROUP to set the audit mode for all members of the group. However, you cannot use AUDIT_MODE to set the audit mode for group members if a user's audit mode is defined in a USER record, XUSER record, or profile group.

AUTHNMTHD

(Informational) Displays the authentication method or methods to be used with the group record; from method 1 to method 32, or none. Used by CA SSO.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIME

Defines the day and time restrictions that govern when an accessor can access a resource.

Use the restrictions parameter with the chres, ch[x]usr, or ch[x]grp commands to modify this property.

The resolution of daytime restrictions is one minute.

EXPIRE_DATE

Defines the date on which an accessor becomes invalid. A value for the EXPIRE_DATE property in a user record overrides a value in a group record.

Note: This property corresponds to the expire[-] parameter of the ch[x]usr and ch[x]grp commands.

FULLNAME

Defines the full name associated with an accessor. CA Access Control uses the full name to identify the accessor in audit log messages, but not for authorization.

FULLNAME is an alphanumeric string. For groups the maximum length is 255 characters. For users the maximum length is 47 characters.

GAPPLS

Defines the list of application groups that the group is authorized to access. Used by CA SSO.

GROUP_MEMBER

Defines the groups that are members of this group.

HOMEDIR

Defines the path of the home directory assigned to a new group member.

Use the homedir parameter with the chgrp, editgrp, or newgrp command to modify this property.

Limit: 255 alphanumeric characters.

INACTIVE

Defines the number of days of inactivity that must pass before the system changes the status of a user to inactive. If the account status is inactive, the user cannot log in.

A value for the INACTIVE property in a USER record overrides a value in a GROUP record. Both override the INACT property in the SEOS class record.

Note: CA Access Control does not store the status; it calculates the status dynamically. To identify inactive users, you must compare the INACTIVE value with the user's LAST_ACC_TIME value.

INACTIVE is part of the profile feature.

MAXLOGINS

Defines the maximum number of concurrent logins that a user is allowed. A zero value indicates that the user can have any number of concurrent logins.

For a user to log in and administer the database (for example, by running selang), MAXLOGINS must be zero or greater than one. This is because CA Access Control considers each task (login, selang, GUI, and so forth) to be a terminal session.

A value for the MAXLOGINS property in a user record overrides a value in a group record. Both override the value of MAXLOGINS in the SEOS class record.

MAXLOGINS is part of the profile feature.

MEMBER_OF

Defines the groups that this group is a member of.

OWNER

Defines the user or group that owns the record.

PASSWDRULES

Specifies the password rules. This property contains a number of fields that determine how CA Access Control handles password protection. For a complete list of the rules, see the modifiable property PROFILE of the USER class.

Use the password parameter and the rules or rules- option with the setoptions command to modify this property.

PASSWDRULES is part of the profile feature.

POLICYMODEL

Specifies the PMDB that receives new passwords when you change user passwords with the sepass utility. The passwords are *not* sent to the Policy Model defined by the parent_pmd or passwd_pmd configuration settings if a value is entered for this property.

Note: This property corresponds to the pmdb[-] parameter of the ch[x]usr and ch[x]grp commands.

POLICYMODEL is part of the profile feature.

PROFUSR

Displays a list of the users associated with this profile group.

PWD_AUTOGEN

Indicates whether the group password is automatically generated. The default is no. Used by CA SSO.

PWD_SYNC

Indicates whether the group password is automatically kept identical for all group applications. The default is no. Used by CA SSO.

PWPOLICY

Defines the record name of the password policy for the group. A password policy is a set of rules for checking the validity of a new password and for defining when a password expires. The default is no validity check. Used by CA SSO.

RESUME_DATE

Defines the date on which a suspended USER account becomes unsuspended.

RESUME_DATE and SUSPEND_DATE work together.

Note: This property corresponds to the resume[-] parameter of the ch[x]usr and ch[x]grp commands.

RESUME_DATE is part of the profile feature.

REVACL

Displays the accessor's access control lists.

SHELL

(UNIX only) The shell program assigned to a new UNIX user when the user is a member of this group.

Use the shellprog parameter with the chxgrp command to modify this property.

SUBGROUP

Displays the list of groups that have this group as a parent.

SUPGROUP

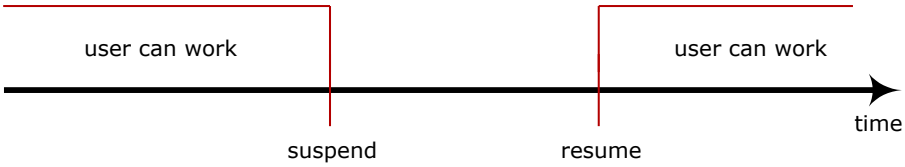
Defines the name of the parent group ("superior" group).

Use the parent[-] parameter with the ch[x]grp command to modify this property.

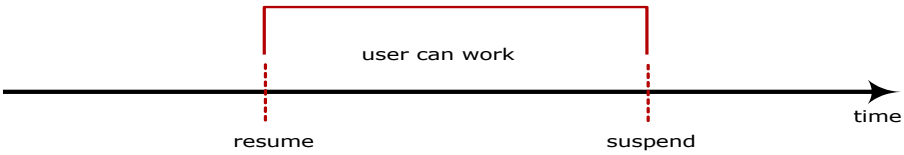
SUSPEND_DATE

Defines the date on which a user account is suspended and so becomes invalid.

If the suspend date for a record precedes its resume date, the user can work before the suspend date and after the resume date.



If a user has a resume date that is earlier than the suspend date, the record is also invalid *before* the resume date. The user can work only between the resume and suspend dates.



A value for the SUSPEND_DATE property in a user record overrides the value in a group record.

Note: This property corresponds to the suspend[-] parameter of the ch[x]usr and ch[x]grp commands.

SUSPEND_WHO

Displays the administrator who activated the suspend date.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

USERLIST

Defines the users that belong to the group.

The user list contained in this property may be different from the one in the native environment USERS property.

Use the join[x][-] commands to modify this property.

GSUDO Class

Each record in the GSUDO class defines a group of actions that Task Delegation-the DO (sesudo)-allows a user to execute or prevents a user from executing. You must create a SUDO class record for each action before adding it to a GSUDO record.

Use GSUDO to define access rules for a group of SUDO resources rather than specifying the same access rule for each resource. You must explicitly connect records of the SUDO class to the GSUDO record in order to group them.

The key of the GSUDO class record is the name of the group.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

ACL

Defines a list of accessors (users and groups) permitted to access the resource, and the accessors' access types.

Each element in the access control list (ACL) contains the following information:

Accessor

Defines an accessor.

Access

Defines the access authority that the accessor has to the resource.

Use the access parameter with the authorize or authorize- command to modify the ACL.

CALACL

Defines a list of the accessors (users and groups) that are permitted to access the resource, and their access types according to the Unicenter NSM calendar status.

Each element in the calendar access control list (CALACL) contains the following information:

Accessor

Defines an accessor.

Calendar

Defines a reference to a calendar in Unicenter TNG.

Access

Defines the access authority that the accessor has to the resource.

Access is permitted only when the calendar is ON. Access is denied in all other cases.

Use the calendar parameter with the authorize command to permit user or group access to the resource according to the access defined in the calendar ACL.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

GROUPS

Defines the list of CONTAINER records that a resource record belongs to.

To modify this property in a class record, change the MEMBERS property in the appropriate CONTAINER record.

Use the mem+ or mem- parameter with the chres, editres or newres command to modify this property.

MEMBERS

The list of objects from the SUDO class that are members of the group.

Use the mem+ or mem- parameter with the chres, editres, and newres commands to modify this property.

NACL

The *NACL* property of a resource is an access control list that defines the accessors that are denied authorization to a resource, together with the type of access that they are denied (for example, write). See also ACL, CALACL, PACL. Each entry in the NACL contains the following information:

Accessor

Defines an accessor.

Access

Defines the type of access that is denied to the accessor.

Use the authorize deniedaccess command, or the authorize- deniedaccess-command, to modify this property.

OWNER

Defines the user or group that owns the record.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

GTERMINAL Class

Each record in the GTERMINAL class defines a group of terminals. You must create a TERMINAL class record for each terminal before adding it to a GTERMINAL record. You must then explicitly connect records of the TERMINAL class to the GTERMINAL record in order to group them.

Terminal groups are useful when defining access rules. You can use a single command to specify an access rule for a group of terminals rather than having to specify the same access rule for each terminal. Similarly, you may apply a rule for a group of terminals by a single command to a group of users.

The key of the GTERMINAL class record is the name of the terminal group.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

ACL

Defines a list of accessors (users and groups) permitted to access the resource, and the accessors' access types.

Each element in the access control list (ACL) contains the following information:

Accessor

Defines an accessor.

Access

Defines the access authority that the accessor has to the resource.

Use the access parameter with the authorize or authorize- command to modify the ACL.

CALACL

Defines a list of the accessors (users and groups) that are permitted to access the resource, and their access types according to the Unicenter NSM calendar status.

Each element in the calendar access control list (CALACL) contains the following information:

Accessor

Defines an accessor.

Calendar

Defines a reference to a calendar in Unicenter TNG.

Access

Defines the access authority that the accessor has to the resource.

Access is permitted only when the calendar is ON. Access is denied in all other cases.

Use the calendar parameter with the authorize command to permit user or group access to the resource according to the access defined in the calendar ACL.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

GROUPS

Defines the list of CONTAINER records that a resource record belongs to.

To modify this property in a class record, change the MEMBERS property in the appropriate CONTAINER record.

Use the mem+ or mem- parameter with the chres, editres or newres command to modify this property.

MEMBERS

The list of objects from the TERMINAL class that are members of the group.

Use the mem+ or mem- parameter with the chres, editres, and newres commands to modify this property.

NACL

The *NACL* property of a resource is an access control list that defines the accessors that are denied authorization to a resource, together with the type of access that they are denied (for example, write). See also ACL, CALACL, PACL. Each entry in the NACL contains the following information:

Accessor

Defines an accessor.

Access

Defines the type of access that is denied to the accessor.

Use the authorize deniedaccess command, or the authorize- deniedaccess-command, to modify this property.

OWNER

Defines the user or group that owns the record.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

GWINSERVICE Class

Each record in the GWINSERVICE class defines a group of Windows Services. Use records in the GWINSERVICE class to define access rules for a group of Windows Services.

The key of a GWINSERVICE class record is the name of the GWINSERVICE record.

ACL

Defines a list of accessors (users and groups) permitted to access the resource, and the accessors' access types.

Each element in the access control list (ACL) contains the following information:

Accessor

Defines an accessor.

Access

Defines the access authority that the accessor has to the resource.

Use the access parameter with the authorize or authorize- command to modify the ACL.

CALACL

Defines a list of the accessors (users and groups) that are permitted to access the resource, and their access types according to the Unicenter NSM calendar status.

Each element in the calendar access control list (CALACL) contains the following information:

Accessor

Defines an accessor.

Calendar

Defines a reference to a calendar in Unicenter TNG.

Access

Defines the access authority that the accessor has to the resource.

Access is permitted only when the calendar is ON. Access is denied in all other cases.

Use the calendar parameter with the authorize command to permit user or group access to the resource according to the access defined in the calendar ACL.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIME

Defines the day and time restrictions that govern when an accessor can access a resource.

Use the restrictions parameter with the chres, ch[x]usr, or ch[x]grp commands to modify this property.

The resolution of daytime restrictions is one minute.

GROUPS

Defines the list of CONTAINER records that a resource record belongs to.

To modify this property in a class record, change the MEMBERS property in the appropriate CONTAINER record.

Use the mem+ or mem- parameter with the chres, editres or newres command to modify this property.

NACL

The *NACL* property of a resource is an access control list that defines the accessors that are denied authorization to a resource, together with the type of access that they are denied (for example, write). See also ACL, CALACL, PACL. Each entry in the NACL contains the following information:

Accessor

Defines an accessor.

Access

Defines the type of access that is denied to the accessor.

Use the authorize deniedaccess command, or the authorize- deniedaccess- command, to modify this property.

NOTIFY

Defines the user to be notified when a resource or user generates an audit event. CA Access Control can email the audit record to the specified user**.**

Limit: 30 characters.

OWNER

Defines the user or group that owns the record.

PACL

Defines a list of accessors that are permitted to access the resource when the access request is made by a specific program (or a program that matches a name-pattern) and their access types. Each element in the program access control list (PACL) contains the following information:

Accessor

Defines an accessor.

Program

Defines a reference to a record in the PROGRAM class, either specifically or by wildcard pattern matching.

Access

Defines the access authority that the accessor has to the resource.

Note: You can use wildcard characters to specify the resource in a PACL.

Use the via(*pgm*) parameter with the selang authorize command to add programs, accessors, and their access types to a PACL, You can use the authorize- command to remove accessors from a PACL.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

HNODE Class

The HNODE class contains information about the organization's CA Access Control hosts. Each record in the class represents a node in the enterprise.

This class is used to manage the information uploaded from the various PMDBs and endpoints and stored on the DMS.

The key of the HNODE class record is the actual host name for an endpoint (for example, myHost.ca.com) or the PMDB name for a Policy Model node (for example, myPMD@myHost.ca.com).

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

ACL

Defines a list of accessors (users and groups) permitted to access the resource, and the accessors' access types.

Each element in the access control list (ACL) contains the following information:

Accessor

Defines an accessor.

Access

Defines the access authority that the accessor has to the resource.

Use the access parameter with the authorize or authorize- command to modify the ACL.

CALACL

Defines a list of the accessors (users and groups) that are permitted to access the resource, and their access types according to the Unicenter NSM calendar status.

Each element in the calendar access control list (CALACL) contains the following information:

Accessor

Defines an accessor.

Calendar

Defines a reference to a calendar in Unicenter TNG.

Access

Defines the access authority that the accessor has to the resource.

Access is permitted only when the calendar is ON. Access is denied in all other cases.

Use the calendar parameter with the authorize command to permit user or group access to the resource according to the access defined in the calendar ACL.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

CATEGORY

Defines one or more security categories assigned to a user or a resource.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

COMPLIANT

Displays the automatically calculated compliance status of the HNODE. Values are:

- yes - CA Access Control installed and all effective policies successfully deployed.
- no - CA Access Control not installed and none of the effective policies were deployed.
- Deviation - CA Access Control installed and not all of the effective policies were deployed.
- Unknown - CA Access Control not installed and there are no effective policies to deploy.

Note: UNAB policies (login and configuration policies) are not assigned with compliant status values.

COMPLIANT_UPDATE_TIME

(Informational) Displays the date when and time when the status changed.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIME

Defines the day and time restrictions that govern when an accessor can access a resource.

Use the restrictions parameter with the chres, ch[x]usr, or ch[x]grp commands to modify this property.

The resolution of daytime restrictions is one minute.

EFFECTIVE_POLICIES

Defines the list of policy versions that should be deployed on this object.

Display name: Effective Policies

GHNODES

Defines the list of host groups this object is a member of.

Display name: node groups

GROUPS

Defines the list of CONTAINER records that a resource record belongs to.

To modify this property in a class record, change the MEMBERS property in the appropriate CONTAINER record.

Use the mem+ or mem- parameter with the chres, editres or newres command to modify this property.

HNODE_IP

The IP address of the host.

Display name: IP

HNODE_KEEP_ALIVE

Defines the last time the HNODE sent a heartbeat to the Distribution Host.

Display name: Last Heartbeat

LOGIN

Defines the default access type to the host.

Display name: LOGIN

NACL

The *NACL* property of a resource is an access control list that defines the accessors that are denied authorization to a resource, together with the type of access that they are denied (for example, write). See also ACL, CALACL, PACL. Each entry in the NACL contains the following information:

Accessor

Defines an accessor.

Access

Defines the type of access that is denied to the accessor.

Use the authorize deniedaccess command, or the authorize- deniedaccess-command, to modify this property.

NODE_INFO

(Informational) Specifies details of the node OS.

NODE_TYPE

(Informational) Defines the type of CA Access Control installation on the host. Valid values are:

- ACU—CA Access Control for UNIX
- ACW—CA Access Control for Windows
- UNAB—UNIX Authentication Broker (UNAB)

Note: A HNODE record can have a value of both ACU and UNAB for the NODE_TYPE property.

NODE_VERSION

(Informational) Defines the CA Access Control version installed on the host. The version number is preceded by the NODE_TYPE.

Example: ACU:12.50-00.647

NOTIFY

Defines the user to be notified when a resource or user generates an audit event. CA Access Control can email the audit record to the specified user**.**

Limit: 30 characters.

OWNER

Defines the user or group that owns the record.

PACL

Defines a list of accessors that are permitted to access the resource when the access request is made by a specific program (or a program that matches a name-pattern) and their access types. Each element in the program access control list (PACL) contains the following information:

Accessor

Defines an accessor.

Program

Defines a reference to a record in the PROGRAM class, either specifically or by wildcard pattern matching.

Access

Defines the access authority that the accessor has to the resource.

Note: You can use wildcard characters to specify the resource in a PACL.

Use the via(*pgm*) parameter with the selang authorize command to add programs, accessors, and their access types to a PACL, You can use the authorize- command to remove accessors from a PACL.

PARENTS

(Informational). The list of PMDBs that are the parents of the node in the propagation tree (also defined by the parent_pmd configuration setting).

POLICYASSIGN

Defines the list of policies that are assigned to this object.

Display name: Assigned Policies

POLICY_STATUS

The status of each of the policies listed in the POLICIES property. The value of the property is a structure with the following fields:

oidPolicy

Object ID of the POLICY object. Same as the value of the POLICIES property.

policy_status

An integer representing one of the following:

- Deployed—Policy was deployed successfully on endpoint.
- Deployed with Failures—Policy was deployed with one or more rules from the deployment script failing to execute on the endpoint.
- Undeployed—Policy was undeployed successfully from endpoint.

  Note: If a policy is undeployed, no status appears for the host (that is, the status is empty).
- Undeployed with Failures—Policy was undeployed with one or more rules from the undeployment script failing to execute on the endpoint.
- Failed Deployment—Policy failed to deploy due to an error in the deployment script.

  Note: This status can appear only if policy verification is enabled. Otherwise, policyfetcher deploys a policy even if the policy contains errors (Deployed with Failures status).
- Unknown—Policy status is unknown.
- Deploy Pending—Waiting for a prerequisite policy to be deployed or the policy contains an undefined or unresolved variable.
- Undeploy Pending—Waiting for a dependent policy to be undeployed.
- Out of Sync—The policy contains a variable and the variable value has changed on the endpoint.
- Not Executed—Policy verification found one or more errors with the policy.
- Queued—Obsolete (for backward compatibility only.)
- Transferred—Obsolete (for backward compatibility only.)
- Transferred Failed—Obsolete (for backward compatibility only.)
- Signature Failed—Obsolete (for backward compatibility only.)

deviation

A value representing whether there is a policy deviation on this node. Valid values are:

- Yes
- No
- Unset

dev_time

Last deviation status update time.

ptime

Last policy status update time.

updator

The name of the user that deployed or removed the policy.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

SECLABEL

Defines the security label of a user or resource.

Note: The SECLABEL property corresponds to the label[-] parameter of the chres and ch[x]usr commands.

SECLEVEL

Defines the security level of an accessor or resource.

Note: This property corresponds to the level[-] parameter of the ch[x]usr and chres commands.

SUBSCRIBER_STATUS

The status of the node per parent. The value of the property is a structure with the following fields:

oidSubs

Object ID of the HNODE object. Same as the value of the SUBSCRIBERS property.

status

A value representing one of the following statuses:

- Available
- Unavailable
- Sync (synchronizing)
- Unknown

stime

Last status update time.

SUBSCRIBERS

The list of subscribers of the node in the propagation tree. Updating this property, implicitly updates the PARENTS property with the value of the HNODE object name.

UACC

Defines the default access authority for the resource, which indicates the access granted to accessors who are not defined to CA Access Control or who do not appear in the ACL of the resource.

Use the defaccess parameter with the chres, editres, or newres command to modify this property.

UNAB_ID

(Informational) Displays the UNAB host ID for reporting purposes.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

HOLIDAY Class

Each record in the HOLIDAY class defines one or more periods when users need extra permission to log in.

Each user has the same access for all the time periods in a record. This means that if you include more than one holiday period in a holiday record, you cannot allow a user to log in during some of those periods and prevent that user from logging in during others. For example, if you want to allow a specific user to log in during New Year's Day but not during Christmas, then the two holidays must be defined in different records.

If you do not specify the year, the holiday is considered annual.

You can override HOLIDAY class restrictions for individual users by specifying the IGN_HOL attribute in the newusr, chusr, or editusr command.

The key of the HOLIDAY class record is the name of the HOLIDAY record.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

ACL

Defines a list of accessors (users and groups) permitted to access the resource, and the accessors' access types.

Each element in the access control list (ACL) contains the following information:

Accessor

Defines an accessor.

Access

Defines the access authority that the accessor has to the resource.

Use the access parameter with the authorize or authorize- command to modify the ACL.

CALACL

Defines a list of the accessors (users and groups) that are permitted to access the resource, and their access types according to the Unicenter NSM calendar status.

Each element in the calendar access control list (CALACL) contains the following information:

Accessor

Defines an accessor.

Calendar

Defines a reference to a calendar in Unicenter TNG.

Access

Defines the access authority that the accessor has to the resource.

Access is permitted only when the calendar is ON. Access is denied in all other cases.

Use the calendar parameter with the authorize command to permit user or group access to the resource according to the access defined in the calendar ACL.

CATEGORY

Defines one or more security categories assigned to a user or a resource.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

GROUPS

Defines the list of CONTAINER records that a resource record belongs to.

To modify this property in a class record, change the MEMBERS property in the appropriate CONTAINER record.

Use the mem+ or mem- parameter with the chres, editres or newres command to modify this property.

HOL_DATE

Specifies the period during which users cannot log in.

The following rules apply to the HOL_DATE property:

- If you do not specify a year, it means the period or holiday is annual. You can specify the year with two digits or four digits, for example: 99 or 1999.
- If you do not specify a start time then the start of the day (midnight) is used; and if you do not specify an end time then the end of the day (midnight) is used.
- If you do not specify an interval of time, but only a date, then the holiday lasts for one whole day.

Use the dates parameter with the chres, editres, and newres commands to modify this property.

NACL

The *NACL* property of a resource is an access control list that defines the accessors that are denied authorization to a resource, together with the type of access that they are denied (for example, write). See also ACL, CALACL, PACL. Each entry in the NACL contains the following information:

Accessor

Defines an accessor.

Access

Defines the type of access that is denied to the accessor.

Use the authorize deniedaccess command, or the authorize- deniedaccess-command, to modify this property.

NOTIFY

Defines the user to be notified when a resource or user generates an audit event. CA Access Control can email the audit record to the specified user**.**

Limit: 30 characters.

OWNER

Defines the user or group that owns the record.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

SECLABEL

Defines the security label of a user or resource.

Note: The SECLABEL property corresponds to the label[-] parameter of the chres and ch[x]usr commands.

SECLEVEL

Defines the security level of an accessor or resource.

Note: This property corresponds to the level[-] parameter of the ch[x]usr and chres commands.

UACC

Defines the default access authority for the resource, which indicates the access granted to accessors who are not defined to CA Access Control or who do not appear in the ACL of the resource.

Use the defaccess parameter with the chres, editres, or newres command to modify this property.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

HOST Class

Each record in the HOST class defines the access that a host has to the local computer when connected by IPv4.

Note: CA Access Control access rules for IP communication apply only to IPv4. CA Access Control does not control access by IPv6.

CA Access Control needs to resolve the addresses of host names that you add to the HOST class. This means the names must appear in the operating system hosts file, or be defined to NIS or DNS.

For each HOST record, the INETACL property defines the services the local host can provide to that host.

CA Access Control permits aliases for a host name, but records that represent aliases are not used for authorization checks. You must know the canonical name of a host for CA Access Control to protect the connection with that host.

The key of the HOST class record is the name of the host.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIME

Defines the day and time restrictions that govern when an accessor can access a resource.

Use the restrictions parameter with the chres, ch[x]usr, or ch[x]grp commands to modify this property.

The resolution of daytime restrictions is one minute.

GROUPS

The list of GHOST or CONTAINER records a resource record belongs to.

To modify this property in a HOST class record, you must change the MEMBERS property in the appropriate CONTAINER or GHOST record.

Use the mem+ or mem- parameter with the chres, editres or newres command to modify this property.

INETACL

Defines the services the local host is allowed to provide to the group of client hosts and what their access types are. Each element in the access control list contains the following information:

Services reference

A reference to a service (a port number or name). To specify all the services, enter an asterisk (*) as the services reference.

CA Access Control supports dynamic port names as specified in the /etc/rpc file (for UNIX) or \etc\rpc file (for Windows).

Access

Defines the access authority that the accessor has to the resource.

Use the access(*type-of-access*), service, and stationName parameters with the authorize[-] command to modify accessors and their access types in the INETACL property.

INSERVRNGE

Specifies the range of services that the local host provides to the group of client hosts.

Performs a similar function to the INETACL property.

Use the service(*serviceRange*) parameter with the authorize[-] command to modify accessors and their access types in the INSERVRANGE property.

OWNER

Defines the user or group that owns the record.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

HOSTNET Class

Each record in the HOSTNET class defines a group of hosts on a particular network. HOSTNET records define rules that govern the access other hosts in the group have to the local host when using IPv4 communication.

Note: CA Access Control access rules for IP communication apply only to IPv4. CA Access Control does not control access by IPv6.

The INMASKMATCH determines which other hosts are subject to a HOSTNET record. The INETACL property defines which services the local host can provide to those hosts.

The key of the HOSTNET class record is the name of the HOSTNET record.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIME

Defines the day and time restrictions that govern when an accessor can access a resource.

Use the restrictions parameter with the chres, ch[x]usr, or ch[x]grp commands to modify this property.

The resolution of daytime restrictions is one minute.

GROUPS

Defines the list of CONTAINER records that a resource record belongs to.

To modify this property in a class record, change the MEMBERS property in the appropriate CONTAINER record.

Use the mem+ or mem- parameter with the chres, editres or newres command to modify this property.

INETACL

Defines the services the local host is allowed to provide to the group of client hosts and what their access types are. Each element in the access control list contains the following information:

Services reference

A reference to a service (a port number or name). To specify all the services, enter an asterisk (*) as the services reference.

CA Access Control supports dynamic port names as specified in the /etc/rpc file (for UNIX) or \etc\rpc file (for Windows).

Access

Defines the access authority that the accessor has to the resource.

Use the access(*type-of-access*), service, and stationName parameters with the authorize[-] command to modify accessors and their access types in the INETACL property.

INSERVRNGE

Specifies the range of services that the local host provides to the group of client hosts.

Performs a similar function to the INETACL property.

Use the service(*serviceRange*) parameter with the authorize[-] command to modify accessors and their access types in the INSERVRANGE property.

INMASKMATCH

Defines the group of hosts to which this HOSTNET record applies. The property contains mask and match values, which are applied to the IP address of the requesting host to determine whether the requesting host belongs to the group.

The INMASKMATCH property only supports addresses that are in IPv4 format .

Note: This property corresponds to the mask and match parameters of the chres command.

OWNER

Defines the user or group that owns the record.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

HOSTNP Class

Each record in the HOSTNP class defines a group of hosts that have similar host names. HOSTNP records define access rules that govern the access other stations (hosts) that match name pattern in the record have to the local host when using IPv4. For each mask (HOSTNP record), the INETACL property lists the service rules that govern the services the local host may provide to the group of hosts.

The key of the HOSTNP class record is the name pattern used to filter the host names of the hosts protected by this HOSTNP record.

Note: CA Access Control access rules for IP communication apply only to IPv4. CA Access Control does not control access by IPv6.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIME

Defines the day and time restrictions that govern when an accessor can access a resource.

Use the restrictions parameter with the chres, ch[x]usr, or ch[x]grp commands to modify this property.

The resolution of daytime restrictions is one minute.

GROUPS

Defines the list of CONTAINER records that a resource record belongs to.

To modify this property in a class record, change the MEMBERS property in the appropriate CONTAINER record.

Use the mem+ or mem- parameter with the chres, editres or newres command to modify this property.

INETACL

Defines the services the local host is allowed to provide to the group of client hosts and what their access types are. Each element in the access control list contains the following information:

Services reference

A reference to a service (a port number or name). To specify all the services, enter an asterisk (*) as the services reference.

CA Access Control supports dynamic port names as specified in the /etc/rpc file (for UNIX) or \etc\rpc file (for Windows).

Access

Defines the access authority that the accessor has to the resource.

Use the access(*type-of-access*), service, and stationName parameters with the authorize[-] command to modify accessors and their access types in the INETACL property.

INSERVRNGE

Specifies the range of services that the local host provides to the group of client hosts.

Performs a similar function to the INETACL property.

Use the service(*serviceRange*) parameter with the authorize[-] command to modify accessors and their access types in the INSERVRANGE property.

OWNER

Defines the user or group that owns the record.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

KMODULE Class

Each record in the KMODULE class defines a kernel module of the operating system.

If a module is defined in the KMODULE class, any call to the operating system to load or unload that module, causes CA Access Control to check the authorizations defined for that module.

The key of a KMODULE record is the name of the kernel module being protected.

Each KMODULE record contains the following properties:

ACL

Defines a list of accessors (users and groups) permitted to access the resource, and the accessors' access types.

Each element in the access control list (ACL) contains the following information:

Accessor

Defines an accessor.

Access

Defines the access authority that the accessor has to the resource.

Use the access parameter with the authorize or authorize- command to modify the ACL.Valid access authorities for KMODULE records are load and unload.

CALACL

Defines a list of the accessors (users and groups) that are permitted to access the resource, and their access types according to the Unicenter NSM calendar status.

Each element in the calendar access control list (CALACL) contains the following information:

Accessor

Defines an accessor.

Calendar

Defines a reference to a calendar in Unicenter TNG.

Access

Defines the access authority that the accessor has to the resource.

Access is permitted only when the calendar is ON. Access is denied in all other cases.

Use the calendar parameter with the authorize command to permit user or group access to the resource according to the access defined in the calendar ACL.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

CATEGORY

Defines one or more security categories assigned to a user or a resource.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIME

Defines the day and time restrictions that govern when an accessor can access a resource.

Use the restrictions parameter with the chres, ch[x]usr, or ch[x]grp commands to modify this property.

The resolution of daytime restrictions is one minute.

FILEPATH

Defines a list of absolute paths to files, each of which contains a kernel module. Separate each file path with a colon (:).

Use more than one file path if you have different versions of the same module.

If no file path is provided, CA Access Control does not perform file path checking on kernel module load.

GROUPS

Defines the list of CONTAINER records that a resource record belongs to.

To modify this property in a class record, change the MEMBERS property in the appropriate CONTAINER record.

Use the mem+ or mem- parameter with the chres, editres or newres command to modify this property.

NACL

The *NACL* property of a resource is an access control list that defines the accessors that are denied authorization to a resource, together with the type of access that they are denied (for example, write). See also ACL, CALACL, PACL. Each entry in the NACL contains the following information:

Accessor

Defines an accessor.

Access

Defines the type of access that is denied to the accessor.

Use the authorize deniedaccess command, or the authorize- deniedaccess- command, to modify this property.

NOTIFY

Defines the user to be notified when a resource or user generates an audit event. CA Access Control can email the audit record to the specified user**.**

Limit: 30 characters.

OWNER

Defines the user or group that owns the record.

PACL

Defines a list of accessors that are permitted to access the resource when the access request is made by a specific program (or a program that matches a name-pattern) and their access types. Each element in the program access control list (PACL) contains the following information:

Accessor

Defines an accessor.

Program

Defines a reference to a record in the PROGRAM class, either specifically or by wildcard pattern matching.

Access

Defines the access authority that the accessor has to the resource.

Note: You can use wildcard characters to specify the resource in a PACL.

Use the via(*pgm*) parameter with the selang authorize command to add programs, accessors, and their access types to a PACL, You can use the authorize- command to remove accessors from a PACL.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

SECLABEL

Defines the security label of a user or resource.

Note: The SECLABEL property corresponds to the label[-] parameter of the chres and ch[x]usr commands.

SECLEVEL

Defines the security level of an accessor or resource.

Note: This property corresponds to the level[-] parameter of the ch[x]usr and chres commands.

SIGNATURE

Displays the unique values for kernel module files defined in the filepath property .

CA Access Control calculates the signatures of kernel modules when it starts up, and when a KMODULE record is changed using selang commands. You can set the signatures yourself using the command seretrust -m.

Note: CA Access Control uses the SIGNATURE property for Linux systems only.

UACC

Defines the default access authority for the resource, which indicates the access granted to accessors who are not defined to CA Access Control or who do not appear in the ACL of the resource.

Use the defaccess parameter with the chres, editres, or newres command to modify this property.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

LOGINAPPL Class

Valid on UNIX

Each record in the LOGINAPPL class defines a login application, identifies who can use the program to log in, and controls the way the login program is used.

The key of the LOGINAPPL class record is the name of the application, that is, a logical name that represents a login application. This logical name is associated, in the LOGINPATH property, with the full path name of the executable.

CA Access Control can also control and protect generic login applications; this means that you can protect groups of login applications that match a certain rule with a generic pattern. To define a generic login application with selang, use the same commands as setting regular login restrictions, except the LOGINPATH parameter, which should include a generic path composed of a regular expression using one or more of the following characters: [,], *, ?.

CA Access Control presets the property values for records in the LOGINAPPL class for standard login programs. You should list and verify the existing settings before making any changes.

Important! LOGINAPPL does not use the _default entry.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

ACL

Defines a list of accessors (users and groups) permitted to access the resource, and the accessors' access types.

Each element in the access control list (ACL) contains the following information:

Accessor

Defines an accessor.

Access

Defines the access authority that the accessor has to the resource.

Use the access parameter with the authorize or authorize- command to modify the ACL.

CALACL

Defines a list of the accessors (users and groups) that are permitted to access the resource, and their access types according to the Unicenter NSM calendar status.

Each element in the calendar access control list (CALACL) contains the following information:

Accessor

Defines an accessor.

Calendar

Defines a reference to a calendar in Unicenter TNG.

Access

Defines the access authority that the accessor has to the resource.

Access is permitted only when the calendar is ON. Access is denied in all other cases.

Use the calendar parameter with the authorize command to permit user or group access to the resource according to the access defined in the calendar ACL.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIME

Defines the day and time restrictions that govern when an accessor can access a resource.

Use the restrictions parameter with the chres, ch[x]usr, or ch[x]grp commands to modify this property.

The resolution of daytime restrictions is one minute.

LOGINFLAGS

Controls special features of the login application, including changes in device number and decrements to the grace logins number. Valid values are:

- **nograce**-Indicates that grace logins should not be decremented when users log in through this application
- **nograceroot**-Indicates that grace logins should not be decremented when root logs in through this application.
- **nologin**-Ensures that a login is entered for the user only. The login is not logged for parent programs.

  A program like rlogin on some platforms causes rlogin to trigger the login and close the login sequence itself; this results in an actual login logged for root. After performing the login, rlogin forks to another program to perform the actual login.

  This problem is apparent if you use a login program such as rlogin or telnet and run seaudit -a. You see that there are also login records for the same login with root as the uid.
- **pamlogin**-Indicates that CA Access Control PAM login interception is used when users log in through this application.

Use the loginflags parameter with the chres, editres, or newres command to modify this property.

LOGINMETHOD

Indicates whether the login application is a pseudo login program for the purposes of CA Access Control protection. Valid values are:

- **normal**-Indicates that this login application executes setuid and setgid calls itself. seosd checks the rules of the specified program.
- **pseudo**-Indicates that this login application calls another program to execute setuid and setgid calls. seosd checks the rules on the other program.

Use the loginmethod parameter with the chres, editres, or newres command to modify this property.

Important! We recommend that you not modify this preset property.

LOGINPATH

The full path (or generic path) to the login application.

Use the loginpath parameter with the chres, editres, or newres command to modify this property.

LOGINSEQUENCE

Defines the sequence of seteuid, setuid, setgid, and setgroups events that seosd processes to set the user from the daemon starting the login process (usually inetd under root) to the user who is actually logged on. You can define up to eight system events.

The login interception sequence always starts with setgid or setgroups events, which are called *triggers*. It ends with a setuid event that changes the user's identity to the real user who logged in.

To successfully accomplish login, the program needs to perform all the specified processes in sequence starting with setgroups or setgid and ending with setuid or seteuid.

Setting the right LoginSequence for a program is a difficult task. Most login programs work well with the default SGRP,SUID setting; this setting means the program issues a setgroups system call and then a setuid command to change the user's identity to the target user.

However, if the SGRP, SUID setting does not work, you must use the following flags to specify the proper order:

- **SEID-**First seteuid event
- **SUID-**First setuid event
- **SGID-**First setgid event
- **SGRP-**First setgroup event
- **FEID-**Second seteuid event
- **FUID-**Second setuid event
- **FGID-**Second setgid event
- **FGRP-**Second setgroup event
- **N3EID-**Third seteuid event
- **N3UID-**Third setuid event
- **N3GID-**Third setgid event
- **N3GRP-**Third setgroup event

Important! You must use the flags to specify the correct login sequence. However, you can specify the flags in any order within the LOGINSEQUENCE parameter. For example, SGRP, SEID, FEID, N3EID is identical to N3EID, FEID, SGRP, SEID.

Note: If you do not know the sequence of system calls that the login program performs, you can view the trace and look for the setuid event that changed the user to the target uid, and then look at prior trace events starting with the first setgid or setgroups event.

For example, if you there is one setgroups event and then only the third setuid call sets the target user, you must set LOGINSEQUENCE to SGRP,SUID,FUID,N3UID. You can specify these flags in any order:

```
SETGRPS : P=565302 to 0,2,3,7,8,10,11,250,220,221,230

SUID  > P=565302 U=0   (R=0   E=0   S=0   ) to (R=0  E=0   S=0   ) () BYPASS
SUID  > P=565302 U=0   (R=0   E=0   S=0   ) to (R=0  E=0   S=-1  ) () BYPASS
LOGIN  : P=565302 User=target Terminal=mercury
```

- The SETGRPS process indicates the trigger.
- The first SUID command should be discounted because you can see that the root simply changed back to root, not the trigger user. (This is the SUID in the sequence.)
- The second SUID command should be discounted as well because you can see that the root changed back to root, not the trigger user. (This is the FUID in the sequence.)
- The LOGIN event is the actual SETUID event causing the login. (Because it is the third event, it is the N3UID flag in the sequence.)

Use the loginsequence parameter with the chres, editres, or newres command to modify this property.

NACL

The *NACL* property of a resource is an access control list that defines the accessors that are denied authorization to a resource, together with the type of access that they are denied (for example, write). See also ACL, CALACL, PACL. Each entry in the NACL contains the following information:

Accessor

Defines an accessor.

Access

Defines the type of access that is denied to the accessor.

Use the authorize deniedaccess command, or the authorize- deniedaccess-command, to modify this property.

NOTIFY

Defines the user to be notified when a resource or user generates an audit event. CA Access Control can email the audit record to the specified user**.**

Limit: 30 characters.

OWNER

Defines the user or group that owns the record.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

UACC

Defines the default access authority for the resource, which indicates the access granted to accessors who are not defined to CA Access Control or who do not appear in the ACL of the resource.

Use the defaccess parameter with the chres, editres, or newres command to modify this property.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

MFTERMINAL Class

Each record in the MFTERMINAL class defines a Mainframe computer that is used to administer CA Access Control. It has the same characteristics as the TERMINAL class, but is not intercepted by CA Access Control.

The key of the MFTERMINAL class is the name of the mainframe computer.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

ACL

Defines a list of accessors (users and groups) permitted to access the resource, and the accessors' access types.

Each element in the access control list (ACL) contains the following information:

Accessor

Defines an accessor.

Access

Defines the access authority that the accessor has to the resource.

Use the access parameter with the authorize or authorize- command to modify the ACL.

CALACL

Defines a list of the accessors (users and groups) that are permitted to access the resource, and their access types according to the Unicenter NSM calendar status.

Each element in the calendar access control list (CALACL) contains the following information:

Accessor

Defines an accessor.

Calendar

Defines a reference to a calendar in Unicenter TNG.

Access

Defines the access authority that the accessor has to the resource.

Access is permitted only when the calendar is ON. Access is denied in all other cases.

Use the calendar parameter with the authorize command to permit user or group access to the resource according to the access defined in the calendar ACL.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

CATEGORY

Defines one or more security categories assigned to a user or a resource.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIME

Defines the day and time restrictions that govern when an accessor can access a resource.

Use the restrictions parameter with the chres, ch[x]usr, or ch[x]grp commands to modify this property.

The resolution of daytime restrictions is one minute.

GROUPS

Defines the list of CONTAINER records that a resource record belongs to.

To modify this property in a class record, change the MEMBERS property in the appropriate CONTAINER record.

Use the mem+ or mem- parameter with the chres, editres or newres command to modify this property.

NACL

The *NACL* property of a resource is an access control list that defines the accessors that are denied authorization to a resource, together with the type of access that they are denied (for example, write). See also ACL, CALACL, PACL. Each entry in the NACL contains the following information:

Accessor

Defines an accessor.

Access

Defines the type of access that is denied to the accessor.

Use the authorize deniedaccess command, or the authorize- deniedaccess- command, to modify this property.

NOTIFY

Defines the user to be notified when a resource or user generates an audit event. CA Access Control can email the audit record to the specified user**.**

Limit: 30 characters.

OWNER

Defines the user or group that owns the record.

PACL

Defines a list of accessors that are permitted to access the resource when the access request is made by a specific program (or a program that matches a name-pattern) and their access types. Each element in the program access control list (PACL) contains the following information:

Accessor

Defines an accessor.

Program

Defines a reference to a record in the PROGRAM class, either specifically or by wildcard pattern matching.

Access

Defines the access authority that the accessor has to the resource.

Note: You can use wildcard characters to specify the resource in a PACL.

Use the via(*pgm*) parameter with the selang authorize command to add programs, accessors, and their access types to a PACL, You can use the authorize- command to remove accessors from a PACL.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

SECLABEL

Defines the security label of a user or resource.

Note: The SECLABEL property corresponds to the label[-] parameter of the chres and ch[x]usr commands.

SECLEVEL

Defines the security level of an accessor or resource.

Note: This property corresponds to the level[-] parameter of the ch[x]usr and chres commands.

UACC

Defines the default access authority for the resource, which indicates the access granted to accessors who are not defined to CA Access Control or who do not appear in the ACL of the resource.

Use the defaccess parameter with the chres, editres, or newres command to modify this property.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

POLICY Class

Each record in the POLICY class defines the information required to deploy and undeploy a policy version. It includes a link to the RULESET objects that contain a list of the selang commands for deploying and undeploying the policy. When the policy is deployed, the deploy selang command is run, which executes all of the commands that define the policy and are stored in the linked RULESET object. When the policy is undeployed, the deploy- selang command is run, which executes all of the commands that refine policy undeployment and are stored in the linked RULESET object.

The key of the POLICY class is the name of the policy followed by the hash symbol (#) and a two-digit version number. For example, mypolicy#13.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

ACL

Defines a list of accessors (users and groups) permitted to access the resource, and the accessors' access types.

Each element in the access control list (ACL) contains the following information:

Accessor

Defines an accessor.

Access

Defines the access authority that the accessor has to the resource.

Use the access parameter with the authorize or authorize- command to modify the ACL.

CALACL

Defines a list of the accessors (users and groups) that are permitted to access the resource, and their access types according to the Unicenter NSM calendar status.

Each element in the calendar access control list (CALACL) contains the following information:

Accessor

Defines an accessor.

Calendar

Defines a reference to a calendar in Unicenter TNG.

Access

Defines the access authority that the accessor has to the resource.

Access is permitted only when the calendar is ON. Access is denied in all other cases.

Use the calendar parameter with the authorize command to permit user or group access to the resource according to the access defined in the calendar ACL.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

CATEGORY

Defines one or more security categories assigned to a user or a resource.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIME

Defines the day and time restrictions that govern when an accessor can access a resource.

Use the restrictions parameter with the chres, ch[x]usr, or ch[x]grp commands to modify this property.

The resolution of daytime restrictions is one minute.

EFFECTS_ON

Defines the list of hosts (HNODE objects) on which this policy is effective (should be deployed).

FINALIZE

Specifies whether this policy version is finalized (can be deployed).

GROUPS

Defines the list of CONTAINER records that a resource record belongs to or the GPOLICY object this policy version belongs to.

Use the mem+ or mem- parameter with the chres, editres or newres command to modify this property.

HNODES

(Informational). The list of CA Access Control nodes which should have this policy deployed.

NACL

The *NACL* property of a resource is an access control list that defines the accessors that are denied authorization to a resource, together with the type of access that they are denied (for example, write). See also ACL, CALACL, PACL. Each entry in the NACL contains the following information:

Accessor

Defines an accessor.

Access

Defines the type of access that is denied to the accessor.

Use the authorize deniedaccess command, or the authorize- deniedaccess- command, to modify this property.

NOTIFY

Defines the user to be notified when a resource or user generates an audit event. CA Access Control can email the audit record to the specified user**.**

Limit: 30 characters.

OWNER

Defines the user or group that owns the record.

PACL

Defines a list of accessors that are permitted to access the resource when the access request is made by a specific program (or a program that matches a name-pattern) and their access types. Each element in the program access control list (PACL) contains the following information:

Accessor

Defines an accessor.

Program

Defines a reference to a record in the PROGRAM class, either specifically or by wildcard pattern matching.

Access

Defines the access authority that the accessor has to the resource.

Note: You can use wildcard characters to specify the resource in a PACL.

Use the via(*pgm*) parameter with the selang authorize command to add programs, accessors, and their access types to a PACL, You can use the authorize- command to remove accessors from a PACL.

POLICY_BASE_NAME

Defines the name of the GPOLICY object this policy version is a member of.

POLICY_VERSION

Defines the version number of this policy version.

POLICY_TYPE

Defines the policy type. Valid values are:

- None
- Login—Specifies that the policy is a UNAB login policy.
- Configuration—Specifies that the policy is a UNAB configuration policy.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

RULESETS

The list of RULESET objects which define the policy.

SECLABEL

Defines the security label of a user or resource.

Note: The SECLABEL property corresponds to the label[-] parameter of the chres and ch[x]usr commands.

SECLEVEL

Defines the security level of an accessor or resource.

Note: This property corresponds to the level[-] parameter of the ch[x]usr and chres commands.

SIGNATURE

A hash value based on signatures of the RULESET objects associated with the policy.

UACC

Defines the default access authority for the resource, which indicates the access granted to accessors who are not defined to CA Access Control or who do not appear in the ACL of the resource.

Use the defaccess parameter with the chres, editres, or newres command to modify this property.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

VARIABLES

(Informational) Displays all versions of the variables in the policy.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

PROCESS Class

Each record in the PROCESS class defines a program-an executable file-that runs in its own address space and that needs to be protected from being killed. Major utilities and database servers are good candidates for such protection since these processes are the main targets for denial-of-service attacks.

Note: When defining a program in the PROCESS class, we recommend that you also define it in the FILE class. This protects the executable by preventing someone from modifying (replacing or corrupting) the executable without authorization.

CA Access Control can protect against three terminate (kill) signals: the regular terminate signal (SIGTERM) and the two signals that an application cannot mask (SIGKILL and SIGSTOP):

Environment	Signal	Number
Windows	KILL	Win32 API
UNIX	Terminate Process	9
UNIX and Windows	STOP	Machine Dependent
UNIX and Windows	TERM	15

Other signals, such as SIGHUP or SIGUSR1, are passed to the process that they target and that process decides whether to ignore the terminate signal or whether to react to it in some way.

The key of the PROCESS class record is the name of the program the record protects. Specify the full path.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Properties marked as *informational* cannot be modified.

ACL

Defines a list of accessors (users and groups) permitted to access the resource, and the accessors' access types.

Each element in the access control list (ACL) contains the following information:

Accessor

Defines an accessor.

Access

Defines the access authority that the accessor has to the resource.

Use the access parameter with the authorize or authorize- command to modify the ACL.

CALACL

Defines a list of the accessors (users and groups) that are permitted to access the resource, and their access types according to the Unicenter NSM calendar status.

Each element in the calendar access control list (CALACL) contains the following information:

Accessor

Defines an accessor.

Calendar

Defines a reference to a calendar in Unicenter TNG.

Access

Defines the access authority that the accessor has to the resource.

Access is permitted only when the calendar is ON. Access is denied in all other cases.

Use the calendar parameter with the authorize command to permit user or group access to the resource according to the access defined in the calendar ACL.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

CATEGORY

Defines one or more security categories assigned to a user or a resource.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIME

Defines the day and time restrictions that govern when an accessor can access a resource.

Use the restrictions parameter with the chres, ch[x]usr, or ch[x]grp commands to modify this property.

The resolution of daytime restrictions is one minute.

GROUPS

Defines the list of CONTAINER records that a resource record belongs to.

To modify this property in a class record, change the MEMBERS property in the appropriate CONTAINER record.

Use the mem+ or mem- parameter with the chres, editres or newres command to modify this property.

NACL

The *NACL* property of a resource is an access control list that defines the accessors that are denied authorization to a resource, together with the type of access that they are denied (for example, write). See also ACL, CALACL, PACL. Each entry in the NACL contains the following information:

Accessor

Defines an accessor.

Access

Defines the type of access that is denied to the accessor.

Use the authorize deniedaccess command, or the authorize- deniedaccess- command, to modify this property.

NOTIFY

Defines the user to be notified when a resource or user generates an audit event. CA Access Control can email the audit record to the specified user**.**

Limit: 30 characters.

OWNER

Defines the user or group that owns the record.

PACL

Defines a list of accessors that are permitted to access the resource when the access request is made by a specific program (or a program that matches a name-pattern) and their access types. Each element in the program access control list (PACL) contains the following information:

Accessor

Defines an accessor.

Program

Defines a reference to a record in the PROGRAM class, either specifically or by wildcard pattern matching.

Access

Defines the access authority that the accessor has to the resource.

Note: You can use wildcard characters to specify the resource in a PACL.

Use the via(*pgm*) parameter with the selang authorize command to add programs, accessors, and their access types to a PACL, You can use the authorize- command to remove accessors from a PACL.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

SECLABEL

Defines the security label of a user or resource.

Note: The SECLABEL property corresponds to the label[-] parameter of the chres and ch[x]usr commands.

SECLEVEL

Defines the security level of an accessor or resource.

Note: This property corresponds to the level[-] parameter of the ch[x]usr and chres commands.

UACC

Defines the default access authority for the resource, which indicates the access granted to accessors who are not defined to CA Access Control or who do not appear in the ACL of the resource.

Use the defaccess parameter with the chres, editres, or newres command to modify this property.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

PROGRAM Class

Each record in the PROGRAM class defines a program that is considered part of the trusted computing base. Programs in this class are trusted not to have security breaches because they are monitored by the Watchdog to ensure they are not modified. If a trusted program is altered, CA Access Control automatically marks the program as untrusted, and the program is prevented from executing. Optionally, you can also allow or prevent execution of untrusted programs using the BLOCKRUN property.

Each PROGRAM record contains several properties that define information about the trusted program file.

Usage notes:

- On UNIX, the PROGRAM class may also contain programs that are not marked as setuid or setgid.
- For non-setuid and non-setgid programs on UNIX, you must also create a record for the trusted executable file in the FILE class in order to authorize the program's execution.

  This lets you track the last accessor information (properties ACCSTIME and ACCSWHO). CA Access Control checks the FILE class first for authorization, and only then checks the PROGRAM class.
- You can define any program as a trusted program within CA Access Control.

  A program cannot be used in a program access control list (PACL) unless it is defined in the PROGRAM class. (However, a program is automatically added to the PROGRAM class when it is added to a PACL.)
- Directories cannot be defined in the PROGRAM class.

The key of the PROGRAM class record is the file name of the program the record protects. You must specify the full path of the file as the object name.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Properties marked *informational* cannot be modified.

ACCSTIME

(Informational). The date and time the record was last accessed.

ACCSWHO

(Informational). The administrator who last accessed the record.

ACL

Defines a list of accessors (users and groups) permitted to access the resource, and the accessors' access types.

Each element in the access control list (ACL) contains the following information:

Accessor

Defines an accessor.

Access

Defines the access authority that the accessor has to the resource.

Use the access parameter with the authorize or authorize- command to modify the ACL.

BLOCKRUN

Specifies whether to check if the program is trusted and blocks the execution of untrusted programs. The execution blocking is performed regardless whether the program is a setuid or a regular program.

Use the blockrun[-] parameter with the chres, editres, and newres commands to modify this property for resources.

CALACL

Defines a list of the accessors (users and groups) that are permitted to access the resource, and their access types according to the Unicenter NSM calendar status.

Each element in the calendar access control list (CALACL) contains the following information:

Accessor

Defines an accessor.

Calendar

Defines a reference to a calendar in Unicenter TNG.

Access

Defines the access authority that the accessor has to the resource.

Access is permitted only when the calendar is ON. Access is denied in all other cases.

Use the calendar parameter with the authorize command to permit user or group access to the resource according to the access defined in the calendar ACL.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

CATEGORY

Defines one or more security categories assigned to a user or a resource.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIME

Defines the day and time restrictions that govern when an accessor can access a resource.

Use the restrictions parameter with the chres, ch[x]usr, or ch[x]grp commands to modify this property.

The resolution of daytime restrictions is one minute.

GROUPS

Defines the list of CONTAINER records that a resource record belongs to.

To modify this property in a class record, change the MEMBERS property in the appropriate CONTAINER record.

Use the mem+ or mem- parameter with the chres, editres or newres command to modify this property.

MD5

(Informational). The RSA-MD5 signature of the file.

NACL

The *NACL* property of a resource is an access control list that defines the accessors that are denied authorization to a resource, together with the type of access that they are denied (for example, write). See also ACL, CALACL, PACL. Each entry in the NACL contains the following information:

Accessor

Defines an accessor.

Access

Defines the type of access that is denied to the accessor.

Use the authorize deniedaccess command, or the authorize- deniedaccess- command, to modify this property.

NOTIFY

Defines the user to be notified when a resource or user generates an audit event. CA Access Control can email the audit record to the specified user**.**

Limit: 30 characters.

OWNER

Defines the user or group that owns the record.

PACL

Defines a list of accessors that are permitted to access the resource when the access request is made by a specific program (or a program that matches a name-pattern) and their access types. Each element in the program access control list (PACL) contains the following information:

Accessor

Defines an accessor.

Program

Defines a reference to a record in the PROGRAM class, either specifically or by wildcard pattern matching.

Access

Defines the access authority that the accessor has to the resource.

Note: You can use wildcard characters to specify the resource in a PACL.

Use the via(*pgm*) parameter with the selang authorize command to add programs, accessors, and their access types to a PACL, You can use the authorize- command to remove accessors from a PACL.

Note: For resources in PROGRAM class, PACL applies only to setuid/setgid programs on UNIX or programs with *file* resource on Windows. CA Access Control first checks for the file resource record, and if the access is allowed, then it checks the program resource record.

PGMINFO

Defines the program information automatically generated by CA Access Control.

The Watchdog automatically verifies the information stored in this property. If it is changed, CA Access Control defines the program as untrusted.

You can select any of the following flags to *exclude* the associated information from this verification process:

crc

The cyclic redundancy check and MD5 signature.

ctime

(UNIX only) The time of the last file status change.

device

On UNIX, the logical disk that the file resides on. On Windows, the drive number of the disk containing the file.

group

The group that owns the program file.

inode

On UNIX, the file system address of the program file. On Windows, this has no meaning

mode

The associated security protection mode for the program file.

mtime

The time the program file was last modified.

owner

The user who owns the program file.

sha1

The SHA1 signature. Digital signature method called Secure Hash Algorithm that could be applied to the program or sensitive files.

size

The size of the program file.

Use the flags, flags+, or flags- parameter with the chres, editres, or newres command to modify the flags in this property.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

SECLABEL

Defines the security label of a user or resource.

Note: The SECLABEL property corresponds to the label[-] parameter of the chres and ch[x]usr commands.

SECLEVEL

Defines the security level of an accessor or resource.

Note: This property corresponds to the level[-] parameter of the ch[x]usr and chres commands.

UACC

Defines the default access authority for the resource, which indicates the access granted to accessors who are not defined to CA Access Control or who do not appear in the ACL of the resource.

Use the defaccess parameter with the chres, editres, or newres command to modify this property.

UNTRUST

Defines whether the resource is untrusted or trusted. If the UNTRUST property is set, accessors cannot use the resource. If the UNTRUST property is not set, the other properties listed in the database for the resource are used to determine accessor's access authority. If a trusted resource is changed in any way, CA Access Control automatically sets the UNTRUST property.

Use the trust[-] parameter with the chres, editres, or newres command to modify this property.

UNTRUSTREASON

(Informational). The reason why the program became untrusted.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

PWPOLICY Class

Each record in the PWPOLICY class defines a password policy. These policies are sets of rules for both the validity of new passwords, and for the length of time the passwords are valid.

The key to the PWPOLICY class is the name of the password policy.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

APPLS

(Informational). The list of CA SSO applications that are linked to the password policy.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

GROUPS

Defines the list of CONTAINER records that a resource record belongs to.

To modify this property in a class record, change the MEMBERS property in the appropriate CONTAINER record.

Use the mem+ or mem- parameter with the chres, editres or newres command to modify this property.

OWNER

Defines the user or group that owns the record.

PASSWDRULES

Specifies the password rules. This property contains a number of fields that determine how CA Access Control handles password protection. For a complete list of the rules, see the modifiable property PROFILE of the USER class.

Use the password parameter and the rules or rules- option with the setoptions command to modify this property.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

REGKEY Class

Valid on Windows

Each record in the REGKEY class defines a key in the Windows registry.

The key to a REGKEY record is the full registry path to the key.

Note: You can use wildcard characters as part of the path specification.

By default CA Access Control protects the CA Access Control registry entries. The root of this registry entry is:

HKEY_LOCAL_MACHINE\SOFTWARE\ComputerAssociates\AccessControl

CA Access Control also protects the following key:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services

The REGKEY class and the REGVAL class have identical properties. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

ACL

Defines a list of accessors (users and groups) permitted to access the resource, and the accessors' access types.

Each element in the access control list (ACL) contains the following information:

Accessor

Defines an accessor.

Access

Defines the access authority that the accessor has to the resource.

Use the access parameter with the authorize or authorize- command to modify the ACL.

CALACL

Defines a list of the accessors (users and groups) that are permitted to access the resource, and their access types according to the Unicenter NSM calendar status.

Each element in the calendar access control list (CALACL) contains the following information:

Accessor

Defines an accessor.

Calendar

Defines a reference to a calendar in Unicenter TNG.

Access

Defines the access authority that the accessor has to the resource.

Access is permitted only when the calendar is ON. Access is denied in all other cases.

Use the calendar parameter with the authorize command to permit user or group access to the resource according to the access defined in the calendar ACL.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIME

Defines the day and time restrictions that govern when an accessor can access a resource.

Use the restrictions parameter with the chres, ch[x]usr, or ch[x]grp commands to modify this property.

The resolution of daytime restrictions is one minute.

GROUPS

Defines the list of CONTAINER records that a resource record belongs to.

To modify this property in a class record, change the MEMBERS property in the appropriate CONTAINER record.

Use the mem+ or mem- parameter with the chres, editres or newres command to modify this property.

NACL

The *NACL* property of a resource is an access control list that defines the accessors that are denied authorization to a resource, together with the type of access that they are denied (for example, write). See also ACL, CALACL, PACL. Each entry in the NACL contains the following information:

Accessor

Defines an accessor.

Access

Defines the type of access that is denied to the accessor.

Use the authorize deniedaccess command, or the authorize- deniedaccess- command, to modify this property.

NOTIFY

Defines the user to be notified when a resource or user generates an audit event. CA Access Control can email the audit record to the specified user**.**

Limit: 30 characters.

OWNER

Defines the user or group that owns the record.

PACL

Defines a list of accessors that are permitted to access the resource when the access request is made by a specific program (or a program that matches a name-pattern) and their access types. Each element in the program access control list (PACL) contains the following information:

Accessor

Defines an accessor.

Program

Defines a reference to a record in the PROGRAM class, either specifically or by wildcard pattern matching.

Access

Defines the access authority that the accessor has to the resource.

Note: You can use wildcard characters to specify the resource in a PACL.

Use the via(*pgm*) parameter with the selang authorize command to add programs, accessors, and their access types to a PACL, You can use the authorize- command to remove accessors from a PACL.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

UACC

Defines the default access authority for the resource, which indicates the access granted to accessors who are not defined to CA Access Control or who do not appear in the ACL of the resource.

Use the defaccess parameter with the chres, editres, or newres command to modify this property.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

REGVAL Class

Valid on Windows

Each record in the REGVAL class defines a value in the Windows registry

The key to a REGVAL record is the full registry path to the value.

Note: You can use wildcard characters as part of the path specification.

The REGVAL class allows the following access types: NONE, READ, WRITE, DELETE.

The REGVAL class and the REGKEY class have identical properties. These properties are as follows. (Non-modifiable properties are marked *informational*.)

ACL

Defines a list of accessors (users and groups) permitted to access the resource, and the accessors' access types.

Each element in the access control list (ACL) contains the following information:

Accessor

Defines an accessor.

Access

Defines the access authority that the accessor has to the resource.

Use the access parameter with the authorize or authorize- command to modify the ACL.

CALACL

Defines a list of the accessors (users and groups) that are permitted to access the resource, and their access types according to the Unicenter NSM calendar status.

Each element in the calendar access control list (CALACL) contains the following information:

Accessor

Defines an accessor.

Calendar

Defines a reference to a calendar in Unicenter TNG.

Access

Defines the access authority that the accessor has to the resource.

Access is permitted only when the calendar is ON. Access is denied in all other cases.

Use the calendar parameter with the authorize command to permit user or group access to the resource according to the access defined in the calendar ACL.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIME

Defines the day and time restrictions that govern when an accessor can access a resource.

Use the restrictions parameter with the chres, ch[x]usr, or ch[x]grp commands to modify this property.

The resolution of daytime restrictions is one minute.

GROUPS

Defines the list of CONTAINER records that a resource record belongs to.

To modify this property in a class record, change the MEMBERS property in the appropriate CONTAINER record.

Use the mem+ or mem- parameter with the chres, editres or newres command to modify this property.

NACL

The *NACL* property of a resource is an access control list that defines the accessors that are denied authorization to a resource, together with the type of access that they are denied (for example, write). See also ACL, CALACL, PACL. Each entry in the NACL contains the following information:

Accessor

Defines an accessor.

Access

Defines the type of access that is denied to the accessor.

Use the authorize deniedaccess command, or the authorize- deniedaccess- command, to modify this property.

NOTIFY

Defines the user to be notified when a resource or user generates an audit event. CA Access Control can email the audit record to the specified user**.**

Limit: 30 characters.

OWNER

Defines the user or group that owns the record.

PACL

Defines a list of accessors that are permitted to access the resource when the access request is made by a specific program (or a program that matches a name-pattern) and their access types. Each element in the program access control list (PACL) contains the following information:

Accessor

Defines an accessor.

Program

Defines a reference to a record in the PROGRAM class, either specifically or by wildcard pattern matching.

Access

Defines the access authority that the accessor has to the resource.

Note: You can use wildcard characters to specify the resource in a PACL.

Use the via(*pgm*) parameter with the selang authorize command to add programs, accessors, and their access types to a PACL, You can use the authorize- command to remove accessors from a PACL.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

UACC

Defines the default access authority for the resource, which indicates the access granted to accessors who are not defined to CA Access Control or who do not appear in the ACL of the resource.

Use the defaccess parameter with the chres, editres, or newres command to modify this property.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

RESOURCE_DESC Class

Each record in the RESOURCE_DESC class defines all of the names that new user-defined class objects are allowed to access in CA SSO. You cannot create a new object in the RESOURCE_DESC class; you can only modify the existing ones.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

CLASS_RIGHT

Of the 32 optional access rights; all are modifiable. The defaults for the first four rights are:

- CLASS_RIGHT1-read
- CLASS_RIGHT2-write
- CLASS_RIGHT3-execute
- CLASS_RIGHT4-rename

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

OWNER

Defines the user or group that owns the record.

RESPONSE_LIST

The name of the object in the RESPONSE_TAB class that contains this object's name.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

RESPONSE_TAB Class

Each record in the RESPONSE_TAB class defines a CA SSO response table to different authorization decisions.

A response is a personalized answer that is returned to application after an authorization request is granted or denied. It consists of KEY=VALUE pairs that are understood by the specific application. The response provides the ability to personalize the portal site according to the user's specific needs and authorization permissions.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

CLASS_RIGHT

32 optional response properties are lists of strings containing KEY=VALUE pairs (for example, button1=yes, picture2=no, and so on). There should be one property for each access value.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

OF_RESOURCE

The name of the object in the RESOURCE_DESC class that refers to the same user-defined class.

OWNER

Defines the user or group that owns the record.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

RULESET Class

Each record in the RULESET class represents a set of rules which define a policy.

The key of the RULESET class record is the name of the policy the record is linked to.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

ACL

Defines a list of accessors (users and groups) permitted to access the resource, and the accessors' access types.

Each element in the access control list (ACL) contains the following information:

Accessor

Defines an accessor.

Access

Defines the access authority that the accessor has to the resource.

Use the access parameter with the authorize or authorize- command to modify the ACL.

CALACL

Defines a list of the accessors (users and groups) that are permitted to access the resource, and their access types according to the Unicenter NSM calendar status.

Each element in the calendar access control list (CALACL) contains the following information:

Accessor

Defines an accessor.

Calendar

Defines a reference to a calendar in Unicenter TNG.

Access

Defines the access authority that the accessor has to the resource.

Access is permitted only when the calendar is ON. Access is denied in all other cases.

Use the calendar parameter with the authorize command to permit user or group access to the resource according to the access defined in the calendar ACL.

CATEGORY

Defines one or more security categories assigned to a user or a resource.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIME

Defines the day and time restrictions that govern when an accessor can access a resource.

Use the restrictions parameter with the chres, ch[x]usr, or ch[x]grp commands to modify this property.

The resolution of daytime restrictions is one minute.

EXPANDED COMMANDS

(Informational) Displays the variable values of the commands in the deployed policy.

EXPANDED UNDO COMMANDS

(Informational) Displays the variable values of the undo commands in the deployed policy.

FINALIZE

Specifies whether the selang scripts have been finalized (and hence the policy version can be deployed).

GROUPS

Defines the list of CONTAINER records that a resource record belongs to.

To modify this property in a class record, change the MEMBERS property in the appropriate CONTAINER record.

Use the mem+ or mem- parameter with the chres, editres or newres command to modify this property.

NACL

The *NACL* property of a resource is an access control list that defines the accessors that are denied authorization to a resource, together with the type of access that they are denied (for example, write). See also ACL, CALACL, PACL. Each entry in the NACL contains the following information:

Accessor

Defines an accessor.

Access

Defines the type of access that is denied to the accessor.

Use the authorize deniedaccess command, or the authorize- deniedaccess-command, to modify this property.

NOTIFY

Defines the user to be notified when a resource or user generates an audit event. CA Access Control can email the audit record to the specified user**.**

Limit: 30 characters.

PACL

Defines a list of accessors that are permitted to access the resource when the access request is made by a specific program (or a program that matches a name-pattern) and their access types. Each element in the program access control list (PACL) contains the following information:

Accessor

Defines an accessor.

Program

Defines a reference to a record in the PROGRAM class, either specifically or by wildcard pattern matching.

Access

Defines the access authority that the accessor has to the resource.

Note: You can use wildcard characters to specify the resource in a PACL.

Use the via(*pgm*) parameter with the selang authorize command to add programs, accessors, and their access types to a PACL, You can use the authorize- command to remove accessors from a PACL.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

RULESET_DOCMD_IDX

(Informational). The command index; that is, a counter of the number of commands in the list of RULESET_DOCMDS.

RULESET_DOCMDS

The list of selang commands which, together, define the policy. These are the commands that are executed to deploy the policy.

Important! Policy deployment does not support commands that set user passwords. Do not include such commands in your deployment script file. UNIX (native) selang commands are supported but will not show in deviation reports.

RULESET_POLICIES

(Informational). The list of policies (POLICY objects) that use this set of rules.

RULESET_UNDOCMD_IDX

(Informational). The command index; that is, a counter of the number of commands in the list of RULESET_UNDOCMDS.

RULESET_UNDOCMDS

The list of selang commands which, together, define the policy undeployment script. These are the commands that are executed to undeploy the policy.

SECLABEL

Defines the security label of a user or resource.

Note: The SECLABEL property corresponds to the label[-] parameter of the chres and ch[x]usr commands.

SECLEVEL

Defines the security level of an accessor or resource.

Note: This property corresponds to the level[-] parameter of the ch[x]usr and chres commands.

SIGNATURE

A hash value based on the RULESET_DOCMDS and RULESET_UNDOCMDS properties.

UACC

Defines the default access authority for the resource, which indicates the access granted to accessors who are not defined to CA Access Control or who do not appear in the ACL of the resource.

Use the defaccess parameter with the chres, editres, or newres command to modify this property.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

SECFILE Class

Each record in the SECFILE class defines a file to be monitored. SECFILE class records provide verification for important files in the system. However, they cannot appear in a conditional access control list.

Add sensitive system files that are not frequently modified to this class to verify that an unauthorized user has not altered them. The following are some examples of the type of files to include in class SECFILE:

For UNIX	For Windows
/.rhosts	\system32\drivers\etc\hosts
/etc/services	\system32\drivers\etc\services
/etc/protocols	\system32\drivers\etc\protocols
/etc/hosts	
/etc/hosts.equiv	

The Watchdog scans these files and ensures the information known about these files is not modified.

Note: Directories cannot be defined in the SECFILE class.

The key of the SECFILE class record is the name of the file that the SECFILE record protects. Specify the full path.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

AIXACL

AIX system ACLs.

AICEXTI

AIX system extended information.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

GROUPS

Defines the list of CONTAINER records that a resource record belongs to.

To modify this property in a class record, change the MEMBERS property in the appropriate CONTAINER record.

Use the mem+ or mem- parameter with the chres, editres or newres command to modify this property.

HPUXACL

HP-UX system ACLs.

MD5

(Informational). The RSA-MD5 signature of the file.

OWNER

Defines the user or group that owns the record.

PGMINFO

Defines the program information automatically generated by CA Access Control.

The Watchdog automatically verifies the information stored in this property. If it is changed, CA Access Control defines the program as untrusted.

You can select any of the following flags to *exclude* the associated information from this verification process:

crc

The cyclic redundancy check and MD5 signature.

ctime

(UNIX only) The time of the last file status change.

device

On UNIX, the logical disk that the file resides on. On Windows, the drive number of the disk containing the file.

group

The group that owns the program file.

inode

On UNIX, the file system address of the program file. On Windows, this has no meaning

mode

The associated security protection mode for the program file.

mtime

The time the program file was last modified.

owner

The user who owns the program file.

sha1

The SHA1 signature. Digital signature method called Secure Hash Algorithm that could be applied to the program or sensitive files.

size

The size of the program file.

Use the flags, flags+, or flags- parameter with the chres, editres, or newres command to modify the flags in this property.

UNTRUST

Defines whether the resource is untrusted or trusted. If the UNTRUST property is set, accessors cannot use the resource. If the UNTRUST property is not set, the other properties listed in the database for the resource are used to determine accessor's access authority. If a trusted resource is changed in any way, CA Access Control automatically sets the UNTRUST property.

Use the trust[-] parameter with the chres, editres, or newres command to modify this property.

UNTRUSTREASON

(Informational). The reason why the program became untrusted.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

SECLABEL Class

Each record in the SECLABEL class associates a security level with security categories. A security label overrides the specific security level and security category assignments in the USER record if the SECLABEL class is active. Assigning a security label is equivalent to explicitly assigning the security level and security categories of the security label to the user.

When a USER record includes a security label, the user is granted access to a resource only if the following conditions are met:

- The user security level specified in the security label is equal to or greater than the resource security level.
- All categories specified in the resource record are included in the security category list of the user security label.

Note: On Windows, each security label defined to CA Access Control must have a record in the SECLABEL class.

The key of the SECLABEL class record is the name of the security label. This name is used to identify the security label when assigning it to a user or resource.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

CATEGORY

Defines one or more security categories assigned to a user or a resource.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

OWNER

Defines the user or group that owns the record.

SECLEVEL

Defines the security level of an accessor or resource.

Note: This property corresponds to the level[-] parameter of the ch[x]usr and chres commands.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

SEOS Class

The SEOS class controls the behavior of the CA Access Control authorization system.

The class contains only one record, called SEOS, which specifies general security and authorization options. To view or change the status of SEOS class properties, use the setoptions command.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

ACCPACL

Indicates the order in which the UACC (defaccess) and PACL lists are scanned during authorization.

When ACCPACL is active and explicit access is provided for a user through an ACL, then that accessor is the allowed access. If there is no explicit access through an ACL but explicit access is defined through a PACL, then the PACL access is the allowed access. If neither ACL nor PACL contains explicit access, defaccess is checked for access definitions.

If ACCPACL is not activated, the ACL is still checked first for explicit access. If the ACL contains no explicit access definitions for the resource being checked, defaccess definitions are checked next. If no explicit access is defined in defaccess, then the PACL access definitions are checked.

When CA Access Control is installed, the value of this property is set to yes.

Use the accpacl or accpacl- parameter with the setoptions command to modify this property.

ADMIN

Indicates whether the ADMIN class is active. Normally the ADMIN class is active and controls permission to perform security administration tasks. If the ADMIN class were inactive, all users could work as CA Access Control administrators.

APPL

Indicates whether the APPL class is active.

AUTHHOST

Indicates whether the AUTHHOST class is active.

CALENDAR

Indicates whether the CALENDAR class is active.

CATEGORY

Indicates whether the CATEGORY class is active.

CNG_ADMIN_PWD

Indicates whether a user with the PWMANAGER attribute can change an ADMIN user password using selang. The default is yes.

Use the class+ or class- parameter and the *cng_adminpwd* option with the setoptions command to activate or inactivate this property.

CNG_OWN_PWD

Indicates whether users can change their own passwords using selang.

Use the class+ or class- parameter and the *cng_ownpwd* option with the setoptions command to activate or inactivate this property.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CONNECT

Indicates whether the CONNECT class is active. When the CONNECT class is active, records in the class protect the outgoing connections.

If the HOST class is active, the CONNECT class is not used as an active class, even when activated.

If the TCP class is active, the CONNECT class is not used as an active class.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIMERES

(UNIX only) Indicates whether CA Access Control checks the daytime restrictions on resources.

DMS

List of DMS servers this database should send notifications to.

DOMAIN

(Windows only) Indicates whether the DOMAIN class is active.

ENDTIME

(Informational). The date and time the database files were last closed in an orderly manner.

FILE

Indicates whether the FILE class is active. When the FILE class is active, records in the class protect files and directories.

ACCGRR

The *accumulative group rights* option (ACCGRR) affects how CA Access Control checks a resource's ACL. If ACCGRR is enabled, CA Access Control checks the ACL for the authorities granted from all the groups to which the user belongs. If ACCGRR is disabled, CA Access Control checks the ACL to see if any of the applicable entries contain the value none. If so, access is denied. Otherwise CA Access Control ignores all group entries except the first applicable one in the access control list.

Use the command setoptions ACCGRR command to enable or disable this property.

HOLIDAY

Indicates whether the HOLIDAY class is active. When the HOLIDAY class is active, users need extra permission to log in during defined Holiday periods.

HOST

Indicates whether the HOST class is active. When the HOST class is active, CA Access Control protects incoming TCP/IP service requests from remote hosts.

If the HOST class is active, the TCP and CONNECT classes are not used as active classes, even when activated.

The default for the HOST class is active.

INACT

Indicates the number of inactive days after which user login is suspended. An inactive day is a day in which the user does not log in.

A value for the INACTIVE property in a USER record overrides a value in a GROUP record. Both override the INACT property in the SEOS class record.

Use the inactive or inactive- parameter with the setoptions command to update this property.

ISDMS

True if the PMDB serves as a DMS.

LOGINAPPL

(UNIX only) Indicates whether the LOGINAPPL class is active.

MAXLOGINS

The maximum number of concurrent logins (terminal sessions) a user is allowed, after which the user is denied access. A zero value indicates no maximum and the user can log in to any number of terminal sessions concurrently. The value must be either zero or greater than 1 if the user wants to log in and run selang or otherwise administer the database, because CA Access Control considers each task (login, selang, GUI, and so forth) to be a terminal session.

A value for the MAXLOGINS property in a USER record overrides a value in a GROUP record. Both override the MAXLOGINS property in the SEOS class record. The value in the SEOS record is the default value used when there is no explicit value in the accessor record.

Use the maxlogins parameter with the chres, editres, and newres commands to modify this property for the SEOS class.

MFTERMINAL

Indicates whether the MFTERMINAL class is active.

PASSWDRULES

Indicates the password rules. This property contains a number of fields that determine how CA Access Control handles password protection. For a complete list of the rules, see the modifiable property PROFILE of the USER class.

Use the password parameter and the rules or rules- option with the setoptions command to modify this property.

PASSWORD

Indicates whether password checking is active.

Use the class+ or class- parameter and the PASSWORD option with the setoptions command to activate or inactivate this property.

PROCESS

Indicates whether the PROCESS class is active. When the PROCESS class is active, records in the class protect defined processes from kill attempts.

The file must also be defined in the FILE class.

PROGRAM

Indicates whether the PROGRAM class is active. When the PROGRAM class is active, records in the class protect defined programs that were marked as Trusted.

PWPOLICY

Indicates whether the PWPOLICY class is active.

REGKEY

(Windows only) Indicates whether the REGKEY class is active.

REGVAL

(Windows only) Indicates whether the REGVAL class is active.

RESOURCE_DESC

Indicates whether the RESOURCE_DESC class is active.

RESPONSE_TAB

Indicates whether the RESPONSE_TAB class is active.

SECLABEL

Indicates whether the SECLABEL class is active.

SECLEVEL

Indicates whether the SECLEVEL class is active.

STARTTIME

(Informational). The date and time the database files were last opened.

SUDO

Indicates whether the SUDO class, used by sesudo, is active.

SYSTEM_AAUDIT_MODE

Specifies the default audit mode (systemwide audit mode) for users and enterprise users.

Default: Failure LoginSuccess LoginFailure

SURROGATE

Indicates whether the SURROGATE class is active. When the SURROGATE class is active, CA Access Control protects surrogate requests.

TCP

Indicates whether the TCP class is active. When the TCP class is active, CA Access Control protects incoming and outgoing TCP services such as mail, ftp, and http.

If the HOST class is active, the TCP class is not used as an active class, even when activated.

If the TCP class is active, the CONNECT class is not used as an active class.

TERMINAL

Indicates whether the TERMINAL class is active. When the TERMINAL class is active, CA Access Control performs a terminal access check during sign-on and protects X-window sessions.

USER_ATTR

Indicates whether the USER_ATTR class is active.

USER_DIR

Indicates whether the USER_DIR class is active.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

SPECIALPGM Class

The SPECIALPGM class gives specified programs special security privileges.

Each record in the SPECIALPGM class has one of two functions:

- Registering backup, DCM, PBF, PBN, STOP, SURROGATE, REGISTRY, and KILL programs in Windows or registering xdm, backup, mail, DCM, PBF, PBN, stop, and surrogate programs in UNIX.
- Associating an application that needs special CA Access Control authorization protection with a logical user ID. This effectively allows setting access permissions according to *what* is being done rather than *who* is doing it.

Note: When defining a program in the SPECIALPGM class, we recommend that you also define it in the FILE class. The FILE resource protects the executable by preventing someone from modifying (replacing or corrupting) the executable without authorization, and the PROGRAM resource ensures that the program does not run if it was modified when CA Access Control was not running

Use the PGMTYPE property to register system services, daemons, or other special programs.

Use the SEOSUID and NATIVEUID properties to assign a logical user to a program.

The key of the SPECIALPGM class record is a path to the special program or to a range, or pattern, of special programs.

Note: The maximum number of rules that you can place in the specialpgm class table is 512.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

NATIVEUID

Indicates the user invoking the program or process. Use * to specify all CA Access Control users.

Use the nativeuid parameter with the chres, editres, or newres command to modify this property.

Note: For backward compatibility with older versions of CA Access Control, you can use the UNIXUID property instead of the NATIVEUID property.

OWNER

Defines the user or group that owns the record.

PGMTYPE

Determines the types of access checks that CA Access Control bypasses when granting access.

backup

Bypasses READ, CHDIR, and UTIME access.

Note: There are two ways to run a successful backup. If the backup program is executed by a non-root user, you have to define this user as an OPERATOR. If the backup program is executed by root, it is enough to register the backup program in the SPECIALPGM class as pgmtype(backup).

changed

(UNIX only) Bypasses PAM enabled surrogate identity change tools, such as su.

For example: *er specialpgm /bin/su pgmtype(changeid)*

dcm

Bypasses all security checks for all events except STOP events.

kill

(Windows only) Bypasses program termination for a process.

For example, the following rule provides a bypass to the services.exe if this process tries to open handles to CA Access Control services (processes) with access mask KILL:

```
nr specialpgm c:\Windows\system32\services.exe pgmtype(kill)
```

On Windows Server 2008, the services.exe process, which manages the stopping and starting of services, opens handles to CA Access Control services (processes) with access type KILL to manage process termination and startup. During installation on Windows Server 2008, CA Access Control runs a discovery process to locate services.exe and creates a bypass rule for it. Without this bypass, you will receive DENIED CA Access Control audit events when services.exe is trying to open handles of CA Access Control services.

mail

(UNIX only) Bypasses database checks for setuid and setgid events. The mail bypass allows you to trace mail access attempts.

none

Removes any PGMTYPE previously set.

pbf

Bypasses database checks for file handling events.

pbn

Bypasses database checks for network-related events.

propagate

(UNIX only) Propagates its own security privileges to any programs that are called from a program with this PGMTYPE. If you do not specify this, SPECIALPGM privileges only affect the parent program.

Note: Security privilege propagation works with PBF, PBN, DCM, and SURROGATE privileges only.

registry

(Windows only) Bypasses database checks for programs that manipulate the Windows registry.

stop

Bypasses database checks for the STOP feature.

surrogate

Bypasses database checks for identity changing events in the kernel. You cannot trace if you use the surrogate bypass.

xdm

(UNIX only) Bypasses network events (such as TCP, HOST, and CONNECT classes) for a limited network range (6000-6010).

Use the pgmtype parameter with the chres, editres, or newres command to modify this property.

SEOSUID

Defines the surrogate logical user authorized to run this special program. This logical user must be defined in the database with a USER record.

Use the seosuid parameter with the chres, editres, or newres command to modify this property.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

Example: Protect a UNIX file

To protect a file that resides in /DATABASE/data/*, the database manager uses a file server daemon called firmdb_filemgr. This file server resides on /opt/dbfirm/bin/firmdb_filemgr. This daemon usually runs under root, making the data accessible to any root-shell hack.

In the following example, the logical user is defined as the only accessor of these files; access by others is restricted:

1. Define the "sensitive" files to CA Access Control using the command:

   ```
   newres file /DATABASE/data/* defaccess(NONE)owner(nobody)
   ```

2. Define the logical user to access the files:

   ```
   newusr firmDB_mgr
   ```

3. Allow only the logical user firmDB_mgr to access the files.

   ```
   authorize file /DATABASE/data/* uid(firmDB_mgr) access(ALL)
   ```

4. Finally, make firmdb_filemgr run with logical user firmDB_mgr

   ```
   newres SPECIALPGM /opt/dbfirm/bin/firmdb_filemgr unixuid(root) \
   seosuid(firmDB_mgr)
   ```

Consequently, when the daemon accesses the files, CA Access Control recognizes the logical user as the accessor of the files, and not root. A hacker who attempts to access the files as root does not succeed.

Example: Protect a Windows file

To protect files that reside in C:\DATABASE\data, the database manager uses a file server service called firmdb_filemgr.exe. This file server resides on C:\Program Files\dbfirm\bin\firmdb_filemgr.exe. This service usually runs under the system account, making the data accessible to any system hack.

In the following example, the logical user is defined as the only accessor of these files; access by others is restricted:

1. Define the "sensitive" files to CA Access Control using the following command:

   ```
   newres file C:\DATABASE\data\* defaccess(NONE)owner(nobody)
   ```

2. Define a logical user to access the files:

   ```
   newusr firmDB_mgr
   ```

3. Allow only the logical user firmDB_mgr to access the files:

   ```
   authorize file C:\DATABASE\data\* uid(firmDB_mgr) access(ALL)
   ```

4. Finally, make firmdb_filemgr run with logical user firmDB_mgr:

   ```
   newres SPECIALPGM ("C:\Program Files\dbfirm\bin\firmdb_filemgr.exe") \
   nativeuid(system) seosuid(firmDB_mgr)
   ```

Consequently, when the service accesses the files, CA Access Control recognizes the logical user as the accessor of the files, and not the system account. A hacker who attempts to access the files in the system account does not succeed.

SUDO Class

Each record in the SUDO class identifies a command for which a user can borrow permissions from another user using the sesudo command.

The key of the SUDO class record is the name of the SUDO record. This name is used instead of the command name when a user executes the commands in the SUDO record.

Note: If you create a SUDO record for an interactive Windows application, you must set the interactive flag for the SUDO record. If you do not set the interactive flag, the application runs in the background and you cannot interact with it. For more information, see the *Troubleshooting Guide*.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

ACL

Defines a list of accessors (users and groups) permitted to access the resource, and the accessors' access types.

Each element in the access control list (ACL) contains the following information:

Accessor

Defines an accessor.

Access

Defines the access authority that the accessor has to the resource.

Use the access parameter with the authorize or authorize- command to modify the ACL.

CALACL

Defines a list of the accessors (users and groups) that are permitted to access the resource, and their access types according to the Unicenter NSM calendar status.

Each element in the calendar access control list (CALACL) contains the following information:

Accessor

Defines an accessor.

Calendar

Defines a reference to a calendar in Unicenter TNG.

Access

Defines the access authority that the accessor has to the resource.

Access is permitted only when the calendar is ON. Access is denied in all other cases.

Use the calendar parameter with the authorize command to permit user or group access to the resource according to the access defined in the calendar ACL.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

CATEGORY

Defines one or more security categories assigned to a user or a resource.

COMMENT

The command that sesudo executes.

The alphanumeric string can contain up to 255 characters, which include the command and also permitted and prohibited parameters.

For example, the following profile definition uses the COMMENT property properly:

newres SUDO *profile_name* comment('command;;NAME')

Note: This use of the COMMENT property is different than in other classes. For more information about defining SUDO records, see the *Endpoint Administration Guide* for your OS. This property was also known as DATA in earlier versions of CA Access Control.

Limit: 255 characters.

Use the comment[-] parameter with the chres, editres, and newres commands to modify this property.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIME

Defines the day and time restrictions that govern when an accessor can access a resource.

Use the restrictions parameter with the chres, ch[x]usr, or ch[x]grp commands to modify this property.

The resolution of daytime restrictions is one minute.

GROUPS

The list of GSUDO or CONTAINER records a resource record belongs to.

To modify this property in a SUDO class record, you must change the MEMBERS property in the appropriate CONTAINER or GSUDO record.

Use the mem+ or mem- parameter with the chres, editres or newres command to modify this property.

INTERACTIVE

(Windows only). This switch should be marked when the application you intend to run via sesudo is an interactive Windows application (for example, notepad.exe or cmd.exe) and not a service application. If you are trying to run an interactive application using sesudo that is not marked as *interactive*, the application runs in the background without the ability to interact with it.

Note: Some Windows applications can not run in the foreground because of a Windows limitation.

NACL

The *NACL* property of a resource is an access control list that defines the accessors that are denied authorization to a resource, together with the type of access that they are denied (for example, write). See also ACL, CALACL, PACL. Each entry in the NACL contains the following information:

Accessor

Defines an accessor.

Access

Defines the type of access that is denied to the accessor.

Use the authorize deniedaccess command, or the authorize- deniedaccess- command, to modify this property.

NOTIFY

Defines the user to be notified when a resource or user generates an audit event. CA Access Control can email the audit record to the specified user**.**

Limit: 30 characters.

OWNER

Defines the user or group that owns the record.

PACL

Defines a list of accessors that are permitted to access the resource when the access request is made by a specific program (or a program that matches a name-pattern) and their access types. Each element in the program access control list (PACL) contains the following information:

Accessor

Defines an accessor.

Program

Defines a reference to a record in the PROGRAM class, either specifically or by wildcard pattern matching.

Access

Defines the access authority that the accessor has to the resource.

Note: You can use wildcard characters to specify the resource in a PACL.

Use the via(*pgm*) parameter with the selang authorize command to add programs, accessors, and their access types to a PACL, You can use the authorize- command to remove accessors from a PACL.

PASSWORDREQ

(UNIX only) Indicates whether the sesudo command requests the original user's password before executing.

Use the password parameter with the chres, editres, or newres command to modify this property.

POLICYMODEL

Specifies the PMDB that receives new passwords when you change user passwords with the sepass utility. The passwords are *not* sent to the Policy Model defined by the parent_pmd or passwd_pmd configuration settings if a value is entered for this property.

Note: This property corresponds to the pmdb[-] parameter of the ch[x]usr and ch[x]grp commands.

SECLABEL

Defines the security label of a user or resource.

Note: The SECLABEL property corresponds to the label[-] parameter of the chres and ch[x]usr commands.

SECLEVEL

Defines the security level of an accessor or resource.

Note: This property corresponds to the level[-] parameter of the ch[x]usr and chres commands.

TARGUSR

(UNIX only) Indicates the target uid, which identifies the user whose permissions are to be borrowed for executing the command. The default is root.

Use the targuid parameter with the chres, editres, or newres command to modify this property.

UACC

Defines the default access authority for the resource, which indicates the access granted to accessors who are not defined to CA Access Control or who do not appear in the ACL of the resource.

Use the defaccess parameter with the chres, editres, or newres command to modify this property.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

SURROGATE Class

Each record in the SURROGATE class defines restrictions that protect a user from other users when they try to change their identity to his or hers. CA Access Control treats a change identity request as an abstract object that can be accessed only by authorized users.

A record in the SURROGATE class represents each user or group who has surrogate protection. Two special records-USER._default and GROUP._default-represent users and groups who do not have individual SURROGATE records. If there is no need to differentiate between the default for users and the default for groups, you may use the _default record for the SURROGATE class instead.

Note: Many Windows utilities and services (for example, Run As) identify as user *NT AUTHORITY\SYSTEM* and not as the original user running them. To let users who use these utilities and services as impersonate another user, you must create this SYSTEM user in the CA Access Control database and authorize it to impersonate the target user.

The key of the SURROGATE class record is the name of the SURROGATE record.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

ACL

Defines a list of accessors (users and groups) permitted to access the resource, and the accessors' access types.

Each element in the access control list (ACL) contains the following information:

Accessor

Defines an accessor.

Access

Defines the access authority that the accessor has to the resource.

Use the access parameter with the authorize or authorize- command to modify the ACL.

CALACL

Defines a list of the accessors (users and groups) that are permitted to access the resource, and their access types according to the Unicenter NSM calendar status.

Each element in the calendar access control list (CALACL) contains the following information:

Accessor

Defines an accessor.

Calendar

Defines a reference to a calendar in Unicenter TNG.

Access

Defines the access authority that the accessor has to the resource.

Access is permitted only when the calendar is ON. Access is denied in all other cases.

Use the calendar parameter with the authorize command to permit user or group access to the resource according to the access defined in the calendar ACL.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

CATEGORY

Defines one or more security categories assigned to a user or a resource.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIME

Defines the day and time restrictions that govern when an accessor can access a resource.

Use the restrictions parameter with the chres, ch[x]usr, or ch[x]grp commands to modify this property.

The resolution of daytime restrictions is one minute.

GROUPS

Defines the list of CONTAINER records that a resource record belongs to.

To modify this property in a class record, change the MEMBERS property in the appropriate CONTAINER record.

Use the mem+ or mem- parameter with the chres, editres or newres command to modify this property.

NACL

The *NACL* property of a resource is an access control list that defines the accessors that are denied authorization to a resource, together with the type of access that they are denied (for example, write). See also ACL, CALACL, PACL. Each entry in the NACL contains the following information:

Accessor

Defines an accessor.

Access

Defines the type of access that is denied to the accessor.

Use the authorize deniedaccess command, or the authorize- deniedaccess- command, to modify this property.

NOTIFY

Defines the user to be notified when a resource or user generates an audit event. CA Access Control can email the audit record to the specified user**.**

Limit: 30 characters.

OWNER

Defines the user or group that owns the record.

PACL

Defines a list of accessors that are permitted to access the resource when the access request is made by a specific program (or a program that matches a name-pattern) and their access types. Each element in the program access control list (PACL) contains the following information:

Accessor

Defines an accessor.

Program

Defines a reference to a record in the PROGRAM class, either specifically or by wildcard pattern matching.

Access

Defines the access authority that the accessor has to the resource.

Note: You can use wildcard characters to specify the resource in a PACL.

Use the via(*pgm*) parameter with the selang authorize command to add programs, accessors, and their access types to a PACL, You can use the authorize- command to remove accessors from a PACL.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

SECLABEL

Defines the security label of a user or resource.

Note: The SECLABEL property corresponds to the label[-] parameter of the chres and ch[x]usr commands.

SECLEVEL

Defines the security level of an accessor or resource.

Note: This property corresponds to the level[-] parameter of the ch[x]usr and chres commands.

UACC

Defines the default access authority for the resource, which indicates the access granted to accessors who are not defined to CA Access Control or who do not appear in the ACL of the resource.

Use the defaccess parameter with the chres, editres, or newres command to modify this property.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

TCP Class

Each record in the TCP class defines a TCP/IP service such as mail, ftp, and http. When the TCP class is being used for authorization, hosts can obtain services from the local host only if the TCP resources grant access. Also, users or groups on a local host can use the TCP/IP services to access remote hosts only if the TCP resources grant access.

The ACL in a TCP record can specify access types for hosts (HOST), groups of hosts (GHOST), networks (HOSTNET), and sets of hosts (HOSTNP).

The CACL in a TCP record can specify access types for hosts (HOST), groups of hosts (GHOST), networks (HOSTNET), and sets of hosts (HOSTNP), and can also specify access types for users and groups.

You can set rules based on IPv4 addresses, not just on host names. This means that you can cater for a domain name change.

Note: CA Access Control access rules for IP communication apply only to IPv4. CA Access Control does not control access by IPv6.

Note: If the CONNECT class is being used as a criterion for access, the TCP class cannot effectively control access. Use either the TCP class or the CONECT class to protect a connection, not both.

The key of the TCP record is the name of the TCP/IP service. The TCP class controls both outgoing services and incoming services.

The following definitions describe the properties contained in a TCP class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational.*

ACL

Defines the hosts for which the local host provides service and the access types that are allowed.

Each element in the access control list contains the following information:

Host reference

Defines a HOST, GHOST, HOSTNET, or HOSTNP record.

Permitted access

The access authority that the referenced host has to the resource. The valid access authorities are:

- **none**—Does not allow the host to perform any operations.
- **read**—Allows the host to obtain TCP service from the local host.

Use the access parameter of the authorize or authorize- command to modify this property

CACL

A list of accessors (users and groups) permitted to access the resource and the host or hosts they can access. Each element in the conditional access control list (CACL) contains the following information:

Accessor

Defines an accessor.

Host reference

Defines a HOST, GHOST, HOSTNET, or HOSTNP record

Access

Defines the access authority that the accessor has to the resource. The valid access types are:

- **write**—Allows the accessor to use this service to access the host or group of hosts.
- **none**—Does not allow the accessor to use this service to access the host or group of hosts.

Use the authorize or authorize- command to modify this property.

CALACL

Defines a list of the accessors (users and groups) that are permitted to access the resource, and their access types according to the Unicenter NSM calendar status.

Each element in the calendar access control list (CALACL) contains the following information:

Accessor

Defines an accessor.

Calendar

Defines a reference to a calendar in Unicenter TNG.

Access

Defines the access authority that the accessor has to the resource.

Access is permitted only when the calendar is ON. Access is denied in all other cases.

Use the calendar parameter with the authorize command to permit user or group access to the resource according to the access defined in the calendar ACL.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIME

Defines the day and time restrictions that govern when an accessor can access a resource.

Use the restrictions parameter with the chres, ch[x]usr, or ch[x]grp commands to modify this property.

The resolution of daytime restrictions is one minute.

GROUPS

Defines the list of CONTAINER records that a resource record belongs to.

To modify this property in a class record, change the MEMBERS property in the appropriate CONTAINER record.

Use the mem+ or mem- parameter with the chres, editres or newres command to modify this property.

NACL

The *NACL* property of a resource is an access control list that defines the accessors that are denied authorization to a resource, together with the type of access that they are denied (for example, write). See also ACL, CALACL, PACL. Each entry in the NACL contains the following information:

Accessor

Defines an accessor.

Access

Defines the type of access that is denied to the accessor.

Use the authorize deniedaccess command, or the authorize- deniedaccess- command, to modify this property.

NOTIFY

Defines the user to be notified when a resource or user generates an audit event. CA Access Control can email the audit record to the specified user**.**

Limit: 30 characters.

OWNER

Defines the user or group that owns the record.

PACL

Defines a list of accessors that are permitted to access the resource when the access request is made by a specific program (or a program that matches a name-pattern) and their access types. Each element in the program access control list (PACL) contains the following information:

Accessor

Defines an accessor.

Program

Defines a reference to a record in the PROGRAM class, either specifically or by wildcard pattern matching.

Access

Defines the access authority that the accessor has to the resource.

Note: You can use wildcard characters to specify the resource in a PACL.

Use the via(*pgm*) parameter with the selang authorize command to add programs, accessors, and their access types to a PACL, You can use the authorize- command to remove accessors from a PACL.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

UACC

Defines the default access authority for the resource, which indicates the access granted to accessors who are not defined to CA Access Control or who do not appear in the ACL of the resource.

Use the defaccess parameter with the chres, editres, or newres command to modify this property.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

TERMINAL Class

Each record in the TERMINAL class defines a terminal of the local host, another host on the network, or an X terminal from which a login session can be made. It can also define terminals that match a terminal name or IP address *pattern* (using wildcards). Terminal permissions are checked during the user login procedure, so that users cannot succeed in logging in from terminals they have not been authorized to use.

The TERMINAL class also controls administrative access. ADMIN users can only administer CA Access Control from terminals for which they have appropriate access permissions.

When you define a new TERMINAL record, CA Access Control tries to convert the name you provide to a fully qualified name. If it succeeds it stores the fully qualified name in the database. If it fails, it stores the name you specified. When you issue subsequent commands referencing this record (chres, showres, rmres, authorize, and so on), you must use the name as it appears in the database.

The key of the TERMINAL record is the name of the terminal. This name identifies the terminal to CA Access Control.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

ACL

Defines a list of accessors (users and groups) permitted to access the resource, and the accessors' access types.

Each element in the access control list (ACL) contains the following information:

Accessor

Defines an accessor.

Access

Defines the access authority that the accessor has to the resource.

Use the access parameter with the authorize or authorize- command to modify the ACL.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

CALACL

Defines a list of the accessors (users and groups) that are permitted to access the resource, and their access types according to the Unicenter NSM calendar status.

Each element in the calendar access control list (CALACL) contains the following information:

Accessor

Defines an accessor.

Calendar

Defines a reference to a calendar in Unicenter TNG.

Access

Defines the access authority that the accessor has to the resource.

Access is permitted only when the calendar is ON. Access is denied in all other cases.

Use the calendar parameter with the authorize command to permit user or group access to the resource according to the access defined in the calendar ACL.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

CATEGORY

Defines one or more security categories assigned to a user or a resource.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIME

Defines the day and time restrictions that govern when an accessor can access a resource.

Use the restrictions parameter with the chres, ch[x]usr, or ch[x]grp commands to modify this property.

The resolution of daytime restrictions is one minute.

GROUPS

The list of GTERMINAL or CONTAINER records a resource record belongs to.

To modify this property in a TERMINAL class record, you must change the MEMBERS property in the appropriate CONTAINER or GTERMINAL record.

Use the mem+ or mem- parameter with the chres, editres or newres command to modify this property.

NACL

The *NACL* property of a resource is an access control list that defines the accessors that are denied authorization to a resource, together with the type of access that they are denied (for example, write). See also ACL, CALACL, PACL. Each entry in the NACL contains the following information:

Accessor

Defines an accessor.

Access

Defines the type of access that is denied to the accessor.

Use the authorize deniedaccess command, or the authorize- deniedaccess- command, to modify this property.

NOTIFY

Defines the user to be notified when a resource or user generates an audit event. CA Access Control can email the audit record to the specified user**.**

Limit: 30 characters.

OWNER

Defines the user or group that owns the record.

PACL

Defines a list of accessors that are permitted to access the resource when the access request is made by a specific program (or a program that matches a name-pattern) and their access types. Each element in the program access control list (PACL) contains the following information:

Accessor

Defines an accessor.

Program

Defines a reference to a record in the PROGRAM class, either specifically or by wildcard pattern matching.

Access

Defines the access authority that the accessor has to the resource.

Note: You can use wildcard characters to specify the resource in a PACL.

Use the via(*pgm*) parameter with the selang authorize command to add programs, accessors, and their access types to a PACL, You can use the authorize- command to remove accessors from a PACL.

SECLABEL

Defines the security label of a user or resource.

Note: The SECLABEL property corresponds to the label[-] parameter of the chres and ch[x]usr commands.

SECLEVEL

Defines the security level of an accessor or resource.

Note: This property corresponds to the level[-] parameter of the ch[x]usr and chres commands.

UACC

Defines the default access authority for the resource, which indicates the access granted to accessors who are not defined to CA Access Control or who do not appear in the ACL of the resource.

Use the defaccess parameter with the chres, editres, or newres command to modify this property.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

UACC Class

Each record in the UACC class defines the default access allowed to a resource class. The UACC record also determines the access level allowed to a resource of that class that is not protected by CA Access Control.

UACC is applicable to most, but not all, classes. The following table shows how each class uses the UACC class.

UACC Usage	Class
Standard	ADMIN, APPL, AUTHHOST, CALENDAR, CONNECT, CONTAINER, DOMAIN, GAPPL, GAUTHHOST, GHOST, GSUDO, GTERMINAL, HOLIDAY, HOST, HOSTNET, HOSTNP, MFTERMINAL, POLICY, PROCESS, PROGRAM, REGKEY, REGVAL, RULESET, SUDO, SURROGATE, TCP, TERMINAL, USER_DIR, User Defined Classes
Nonstandard	FILE, GFILE
None	AGENT, AGENT_TYPE, CATEGORY, GROUP, PWPOLICY, RESOURCE_DESC, RESPONSE_TAB, SECFILE, SECLABEL, SEOS, SPECIALPGM, USER, USER_ATTR

For users outside the special _restricted group, the record for FILE in the UACC class protects only files that are part of CA Access Control-such as the seos.ini, seosd.trace, seos.audit, and seos.error files. These files are not explicitly defined to CA Access Control, but are automatically protected by CA Access Control.

The key of the UACC class record is the name of the class whose UACC properties are being defined.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

ACL

Defines a list of accessors (users and groups) permitted to access the resource, and the accessors' access types.

Each element in the access control list (ACL) contains the following information:

Accessor

Defines an accessor.

Access

Defines the access authority that the accessor has to the resource.

Use the access parameter with the authorize or authorize- command to modify the ACL.

ALLOWACCS

A list of all allowed accesses for this class.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

CALACL

Defines a list of the accessors (users and groups) that are permitted to access the resource, and their access types according to the Unicenter NSM calendar status.

Each element in the calendar access control list (CALACL) contains the following information:

Accessor

Defines an accessor.

Calendar

Defines a reference to a calendar in Unicenter TNG.

Access

Defines the access authority that the accessor has to the resource.

Access is permitted only when the calendar is ON. Access is denied in all other cases.

Use the calendar parameter with the authorize command to permit user or group access to the resource according to the access defined in the calendar ACL.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

NACL

The *NACL* property of a resource is an access control list that defines the accessors that are denied authorization to a resource, together with the type of access that they are denied (for example, write). See also ACL, CALACL, PACL. Each entry in the NACL contains the following information:

Accessor

Defines an accessor.

Access

Defines the type of access that is denied to the accessor.

Use the authorize deniedaccess command, or the authorize- deniedaccess-command, to modify this property.

OWNER

Defines the user or group that owns the record.

UACC

Defines the default access authority for the resource, which indicates the access granted to accessors who are not defined to CA Access Control or who do not appear in the ACL of the resource.

Use the defaccess parameter with the chres, editres, or newres command to modify this property.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

USER Class

Each record in the USER class defines a user in the CA Access Control database.

The key of the USER record is the name of the user—the name entered by the user when logging into the system.

You can change most of the USER properties from the CA Access Control Endpoint Management, or by using the selang command chusr. Properties you cannot change using chusr are labeled *informational.*

Note: In most cases, and unless otherwise indicated, to change a property using chusr, you use the property name as the command parameter.

You can view all properties from the CA Access Control Endpoint Management or by using the selang command showusr.

APPLIST

Used by CA SSO.

APPLIST_TIME

Used by CA SSO.

APPLS

(Informational) Displays the list of applications that the accessor is authorized to access. Used by CA SSO.

AUDIT_MODE

Defines the activities that CA Access Control records in the audit log. You can specify any combination of the following activities:

- No logging
- All activities recorded in the trace file
- Unsuccessful login attempts
- Successful logins
- Failed access attempts to resources protected by CA Access Control
- Successful accesses to resources protected by CA Access Control

Note: This property corresponds to the audit parameter of the ch[x]usr and ch[x]grp commands.

AUTHNMTHD

(Informational) Displays the authentication method or methods to be used with the group record; from method 1 to method 32, or none. Used by CA SSO.

BADPASSWD

Used by CA SSO.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

CATEGORY

Defines one or more security categories assigned to a user or a resource.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

COUNTRY

A string that specifies a country descriptor for a user. This string is part of the X.500 naming scheme. CA Access Control does not use it for authorization.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIME

Defines the day and time restrictions that govern when an accessor can access a resource.

Use the restrictions parameter with the chres, ch[x]usr, or ch[x]grp commands to modify this property.

The resolution of daytime restrictions is one minute.

EMAIL

Defines the email address of the user, up to 128 characters.

EXPIRE_DATE

Defines the date on which an accessor becomes invalid. A value for the EXPIRE_DATE property in a user record overrides a value in a group record.

Note: This property corresponds to the expire[-] parameter of the ch[x]usr and ch[x]grp commands.

FULLNAME

Defines the full name associated with an accessor. CA Access Control uses the full name to identify the accessor in audit log messages, but not for authorization.

FULLNAME is an alphanumeric string. For groups the maximum length is 255 characters. For users the maximum length is 47 characters.

GAPPLS

(Informational) Indicates the list of application groups that the user is authorized to access. Used by CA SSO.

GRACELOGIN

Defines the number of grace logins a user has after a password expires. When the number of grace logins is exceeded, the user is denied access to the system and must contact the system administrator for a new password.

The number of grace logins must be between 0 and 255. If this value is 0, the user cannot log in.

A value for the GRACELOGIN property in a USER record overrides a value for NGRACE in a GROUP record. Both override the PASSWDRULES property in the SEOS class record.

Note: This property corresponds to the grace parameter of the ch[x]usr command.

GROUPS

(Informational) Displays the list of user groups that the user belongs to. This property also contains any group authorities, such as group administration authority (GROUP-ADMIN), assigned to the user for each group the user belongs to.

The group list contained in this property may be different from the one in the native environment GROUPS property.

Note: This property is not modified by the ch[x]usr command. Instead, use the join[-] or joinx[-]command to modify this property.

HOMEDIR

(UNIX only) Defines the user's home directory. Used by CA SSO.

INACTIVE

Defines the number of days of inactivity that must pass before the system changes the status of a user to inactive. If the account status is inactive, the user cannot log in.

A value for the INACTIVE property in a USER record overrides a value in a GROUP record. Both override the INACT property in the SEOS class record.

Note: CA Access Control does not store the status; it calculates the status dynamically. To identify inactive users, you must compare the INACTIVE value with the user's LAST_ACC_TIME value.

LAST_ACC_TERM

Displays the terminal from which the last login was performed.

LAST_ACC_TIME

Displays the date and time of the last login.

LOCALAPPS

Used by CA SSO.

LOCATION

Defines a user location. CA Access Control does not use this information for authorization.

LOGININFO

Defines the information needed to log the user into a specific application and audit data. LOGININFO contains a separate list for each application that the user is authorized to access. Used by CA SSO.

LOGSHIFT

Indicates whether a login outside of the shift time frame is permitted. CA Access Control writes an audit record in the audit log for this event.

MAXLOGINS

Defines the maximum number of concurrent logins that a user is allowed. A zero value indicates that the user can have any number of concurrent logins.

For a user to log in and administer the database (for example, by running selang), MAXLOGINS must be zero or greater than one. This is because CA Access Control considers each task (login, selang, GUI, and so forth) to be a terminal session.

A value for the MAXLOGINS property in a user record overrides a value in a group record. Both override the value of MAXLOGINS in the SEOS class record.

MIN_TIME

Defines the minimum time in days allowed between password changes for the user.

A value for the MIN_TIME property in a USER record overrides a value in a GROUP record. Both override the PASSWDRULES property in the SEOS class record.

Note: This property corresponds to the min_life parameter of the ch[x]usr command.

NOTIFY

Defines the user to be notified when a resource or user generates an audit event. CA Access Control can email the audit record to the specified user**.**

Limit: 30 characters.

OBJ_TYPE

Specifies the user authority attributes. Each of these attributes corresponds to the parameter of the same name in the ch[x]usr command. A user can have one or more of the following authority attributes:

ADMIN

Specifies whether the user can perform administrative functions, similar to root in the UNIX environment.

AUDITOR

Specifies whether the user can monitor the system, list information in the database, and set the audit mode for existing records.

IGN_HOL

Specifies whether the user can log in during any period of time defined in a HOLIDAY record.

LOGICAL

Specifies that the user is only for internal CA Access Control purposes and cannot be used by a real user to log in.

For example, the user nobody that you can use as the owner of resources to prevent even the resource owner from accessing the resource is a logical user by default. This means that no user can log in using this account.

OPERATOR

Specifies whether the user can list everything in the database and use the secons utility.

PWMANAGER

Specifies whether the user can modify the password settings of other users and can enable a user account that has been disabled by the serevu utility.

SERVER

Specifies whether a process can ask users for authorization and can issue the SEOSROUTE_VerifyCreate API call.

OIDCRDDATA

Used by CA SSO.

OLD_PASSWD

Contains an encrypted list of the user's previous passwords. The user cannot choose a new password from this list. The maximum number of passwords saved in OLD_PASSWD is determined by the setoptions command.

ORG_UNIT

A string that stores information on the organizational unit in which the user works. This string is part of the X.500 naming scheme. CA Access Control does not use it for authorization.

ORGANIZATION

Defines the organization in which the user works. This string is part of the X.500 naming scheme. CA Access Control does not use this for authorization.

OWNER

Defines the user or group that owns the record.

PASSWD_A_C_W

Indicates the ADMIN user who last changed the user password for this record.

PASSWD_INT

Defines the maximum time in days between password changes for users.

A value for the PASSWD_INT property in a USER record overrides the value in a GROUP record. Both override the PASSWDRULES property in the SEOS class record.

Note: This property corresponds to the interval parameter of the ch[x]usr command.

PASSWD_L_A_C

Displays the date and time at which an administrator last updated the password.

PASSWD_L_C

Displays the date and time at which a user last updated the password.

PGMINFO

Defines the program information automatically generated by CA Access Control.

The Watchdog automatically verifies the information stored in this property. If it is changed, CA Access Control defines the program as untrusted.

You can select any of the following flags to *exclude* the associated information from this verification process:

crc

The cyclic redundancy check and MD5 signature.

ctime

(UNIX only) The time of the last file status change.

device

On UNIX, the logical disk that the file resides on. On Windows, the drive number of the disk containing the file.

group

The group that owns the program file.

inode

On UNIX, the file system address of the program file. On Windows, this has no meaning

mode

The associated security protection mode for the program file.

mtime

The time the program file was last modified.

owner

The user who owns the program file.

sha1

The SHA1 signature. Digital signature method called Secure Hash Algorithm that could be applied to the program or sensitive files.

size

The size of the program file.

Use the flags, flags+, or flags- parameter with the chres, editres, or newres command to modify the flags in this property.

PHONE

Defines the user's telephone number. This information is not used for authorization.

POLICYMODEL

Specifies the PMDB that receives new passwords when you change user passwords with the sepass utility. The passwords are *not* sent to the Policy Model defined by the parent_pmd or passwd_pmd configuration settings if a value is entered for this property.

Note: This property corresponds to the pmdb[-] parameter of the ch[x]usr and ch[x]grp commands.

PROFILE

Defines the path to the user's profile. This string can include a local absolute path, or a UNC path.

PWD_AUTOGEN

Displays whether the user password is automatically generated. Used by CA SSO.

The default is no.

PWD_SYNC

Displays whether the user password is automatically kept identical for all user applications. Used by CA SSO.

The default is no.

RESUME_DATE

Defines the date on which a suspended USER account becomes unsuspended.

RESUME_DATE and SUSPEND_DATE work together.

Note: This property corresponds to the resume[-] parameter of the ch[x]usr and ch[x]grp commands.

REVACL

Displays the access control lists of the accessor.

REVOKE_COUNT

Used by CA SSO.

SCRIPT_VARS

Used by CA SSO, Defines a variables list with the variable values of the application script that are saved per application.

SECLABEL

Defines the security label of a user or resource.

Note: The SECLABEL property corresponds to the label[-] parameter of the chres and ch[x]usr commands.

SECLEVEL

Defines the security level of an accessor or resource.

Note: This property corresponds to the level[-] parameter of the ch[x]usr and chres commands.

SESSION_GROUP

Defines an SSO session group for a user. The SESSION_GROUP property is a string with a maximum length of 16 characters.

In Windows, an administrator can enter a session group new name if the preferred name is not in the drop-down list.

Used by CA SSO.

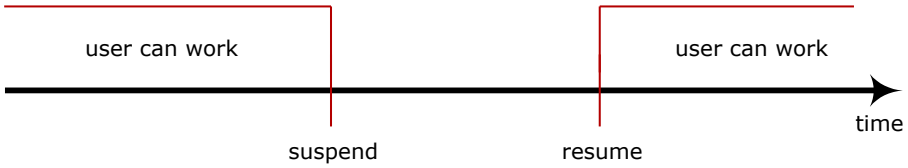
SHIFT

Used by CA SSO.

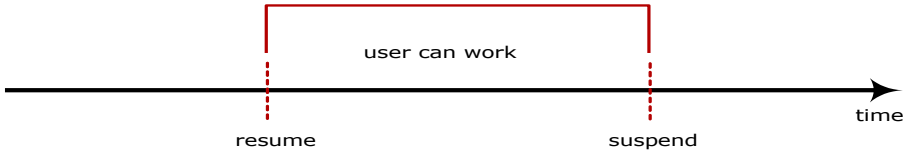
SUSPEND_DATE

Defines the date on which a user account is suspended and so becomes invalid.

If the suspend date for a record precedes its resume date, the user can work before the suspend date and after the resume date.



If a user has a resume date that is earlier than the suspend date, the record is also invalid *before* the resume date. The user can work only between the resume and suspend dates.



A value for the SUSPEND_DATE property in a user record overrides the value in a group record.

Note: This property corresponds to the suspend[-] parameter of the ch[x]usr and ch[x]grp commands.

SUSPEND_WHO

Displays the administrator who activated the suspend date.

Note: This property corresponds to the suspend[-] parameter of the ch[x]usr command.

UALIAS

Displays the aliases of a specific user defined to one or more authentication hosts. Used by CA SSO.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

USER_ATTR Class

Each record in the USER_ATTR class defines the valid user attributes of a CA SSO user directory.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

ATTR_PREDEFS

The list of allowed values for a specific attribute.

ATTRNAME

(Informational). The name of the attribute.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DBFIELD

The name of the field in the userdir database. Since different databases can contain different attributes, the attribute fields should be synchronized.

FIELDID

(Informational). The ID of the DB field

OWNER

Defines the user or group that owns the record.

PARAMETER_TYPE

Indicates whether the user attribute is a string or numeric.

PRIORITY

The priority of the user attribute: when setting an authorization rule to a PARAM_RULE object (such as APPL, URL) the rule is defined with the priority that the user attribute refers to.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

USER_DIR_PROP

(Informational). The name of the user's directory.

USERATTR_FLAGS

Contains information about the attribute. The flag can contain the following values:

- **aznchk-**Indicates whether to use this attribute for authorization.
- **predef** (predefined), **freetex** (free text), or **userdir** (user directory)-These three values specify the source of the user attributes.
- **user** or **group-**These values indicate whether the attribute (accessor) is a user or a group.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

USER_DIR Class

Each record in the USER_DIR class defines a CA SSO user directory.

The key of the USER_DIR record is the name of the directory.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

ADMIN_NAME

Login name of the administrator of the directory.

ADMIN_PWD

Password of the administrator of the directory. The password is stored in clear text format. It is not displayed in selang but can be obtained with seadmapi functions.

AZNACL

Defines the authorization ACL. The authorization ACL is an ACL that allows access to a resource based on the resource description. The description is sent to the authorization engine, not the object. Typically, when an AZNACL is used, the object is not in the database.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CONTOBJ_CLS

The names of the classes the container object inherits from (needed for creation of new login info containers in LDAP.)

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DIR_TYPE

The type of directory. Valid values are: ETRUST_AC, LDAP, ODBC, NT_Domain or none.

GRPOBJ_CLS

The names of the classes the group object inherits from (needed for creation of new groups in LDAP.)

LICONTOBJ_CLS

The names of the classes the login info container object inherits from (needed for creation of new login info containers in LDAP.)

LIOBJ_CLS

The names of the classes the login info object inherits from (needed for creation of new login information in LDAP.)

MAX_RET_ITEMS

The maximum number of items retrieved. The default depends on the directory type.

OWNER

Defines the user or group that owns the record.

PATH

The relative distinguishing name in the LDAP tree to begin all queries.

PORT_NUM

The port number on the host computer used to access the directory.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

TIMEOUT_CON

The time (in seconds) the system waits to connect to the directory before issuing a Timeout error message.

UACC

Defines the default access authority for the resource, which indicates the access granted to accessors who are not defined to CA Access Control or who do not appear in the ACL of the resource.

Use the defaccess parameter with the chres, editres, or newres command to modify this property.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

USERATTR_LIST

The list of objects in the USER_ATTR class that was created with this USER_DIR object as the value for the USER_DIR parameter.

USERDIR_HOST

The name of the host computer for the directory. This property must be defined in the class record.

USROBJ_CLS

The names of the classes the user object inherits from (needed for creation of new users in LDAP.)

VERSION

The version number of the directory.

WEBSERVICE Class

The WEBSERVICE class is obsolete; it is not used by CA Access Control.

WINSERVICE Class

Each record in the WINSERVICE class defines a Windows Service. Use records in the WINSERVICE class to define access rules for Windows Services.

The key of a WINSERVICE class record is the Windows name of the service.

Note: In most cases, and unless otherwise indicated, to change a property using the selang chres command, you use the property name as the command parameter.

You can view all properties from the CA Access Control Endpoint Management or by using the selang command showres WINSERVICE.

ACL

Defines a list of accessors (users and groups) permitted to access the resource, and the accessors' access types.

Each element in the access control list (ACL) contains the following information:

Accessor

Defines an accessor.

Access

Defines the access authority that the accessor has to the resource.

Use the access parameter with the authorize or authorize- command to modify the ACL.

CALACL

Defines a list of the accessors (users and groups) that are permitted to access the resource, and their access types according to the Unicenter NSM calendar status.

Each element in the calendar access control list (CALACL) contains the following information:

Accessor

Defines an accessor.

Calendar

Defines a reference to a calendar in Unicenter TNG.

Access

Defines the access authority that the accessor has to the resource.

Access is permitted only when the calendar is ON. Access is denied in all other cases.

Use the calendar parameter with the authorize command to permit user or group access to the resource according to the access defined in the calendar ACL.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

CATEGORY

Defines one or more security categories assigned to a user or a resource.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIME

Defines the day and time restrictions that govern when an accessor can access a resource.

Use the restrictions parameter with the chres, ch[x]usr, or ch[x]grp commands to modify this property.

The resolution of daytime restrictions is one minute.

GROUPS

Defines the list of CONTAINER records that a resource record belongs to.

To modify this property in a class record, change the MEMBERS property in the appropriate CONTAINER record.

Use the mem+ or mem- parameter with the chres, editres or newres command to modify this property.

NACL

The *NACL* property of a resource is an access control list that defines the accessors that are denied authorization to a resource, together with the type of access that they are denied (for example, write). See also ACL, CALACL, PACL. Each entry in the NACL contains the following information:

Accessor

Defines an accessor.

Access

Defines the type of access that is denied to the accessor.

Use the authorize deniedaccess command, or the authorize- deniedaccess- command, to modify this property.

NOTIFY

Defines the user to be notified when a resource or user generates an audit event. CA Access Control can email the audit record to the specified user**.**

Limit: 30 characters.

OWNER

Defines the user or group that owns the record.

PACL

Defines a list of accessors that are permitted to access the resource when the access request is made by a specific program (or a program that matches a name-pattern) and their access types. Each element in the program access control list (PACL) contains the following information:

Accessor

Defines an accessor.

Program

Defines a reference to a record in the PROGRAM class, either specifically or by wildcard pattern matching.

Access

Defines the access authority that the accessor has to the resource.

Note: You can use wildcard characters to specify the resource in a PACL.

Use the via(*pgm*) parameter with the selang authorize command to add programs, accessors, and their access types to a PACL, You can use the authorize- command to remove accessors from a PACL.

RAUDIT

Defines the types of access events that CA Access Control records in the audit log. RAUDIT derives its name from *R*esource *AUDIT*. Valid values are:

all

All access requests.

success

Granted access requests.

failure

Denied access requests (default).

none

No access requests.

CA Access Control records events on each attempted access to a resource, and does not record whether the access rules were applied directly to the resource, or were applied to a group or class that had the resource as a member.

Use the audit parameter of the chres and chfile commands to modify the audit mode.

SECLABEL

Defines the security label of a user or resource.

Note: The SECLABEL property corresponds to the label[-] parameter of the chres and ch[x]usr commands.

SECLEVEL

Defines the security level of an accessor or resource.

Note: This property corresponds to the level[-] parameter of the ch[x]usr and chres commands.

UACC

Defines the default access authority for the resource, which indicates the access granted to accessors who are not defined to CA Access Control or who do not appear in the ACL of the resource.

Use the defaccess parameter with the chres, editres, or newres command to modify this property.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

WARNING

Specifies whether Warning mode is enabled. When Warning mode is enabled on a resource, all access requests to the resource are granted, and if an access request violates an access rule, a record is written to the audit log.

XGROUP Class

Each record in the XGROUP class defines a group of users in the database.

The key of each XGROUP class record is the name of the group.

Note: The properties of profile groups apply to each user associated with the profile group. However, if the same property is specified in a user (USER or XUSER) record, the user record overrides those in the profile group record.

You can change most of these properties from the CA Access Control Endpoint Management, or by using the selang command chxgrp.

Note: In most cases, and unless otherwise indicated, to change a property using chxgrp, you use the property name as the command parameter.

You can view all properties from the CA Access Control Endpoint Management, or by using the selang command showxgrp.

APPLS

(Informational) Displays the list of applications that the accessor is authorized to access. Used by CA SSO.

AUDIT_MODE

Defines the activities that CA Access Control records in the audit log. You can specify any combination of the following activities:

- No logging
- All activities recorded in the trace file
- Unsuccessful login attempts
- Successful logins
- Failed access attempts to resources protected by CA Access Control
- Successful accesses to resources protected by CA Access Control

Note: This property corresponds to the audit parameter of the ch[x]usr and ch[x]grp commands. You can use AUDIT_MODE for a GROUP or XGROUP to set the audit mode for all members of the group. However, you cannot use AUDIT_MODE to set the audit mode for group members if a user's audit mode is defined in a USER record, XUSER record, or profile group.

AUTHNMTHD

(Informational) Displays the authentication method or methods to be used with the group record; from method 1 to method 32, or none. Used by CA SSO.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIME

Defines the day and time restrictions that govern when an accessor can access a resource.

Use the restrictions parameter with the chres, ch[x]usr, or ch[x]grp commands to modify this property.

The resolution of daytime restrictions is one minute.

EXPIRE_DATE

Defines the date on which an accessor becomes invalid. A value for the EXPIRE_DATE property in a user record overrides a value in a group record.

Note: This property corresponds to the expire[-] parameter of the ch[x]usr and ch[x]grp commands.

FULLNAME

Defines the full name associated with an accessor. CA Access Control uses the full name to identify the accessor in audit log messages, but not for authorization.

FULLNAME is an alphanumeric string. For groups the maximum length is 255 characters. For users the maximum length is 47 characters.

GAPPLS

Defines the list of application groups that the group is authorized to access. Used by CA SSO.

GROUP_MEMBER

Defines the groups that are members of this group.

MEMBER_OF

Defines the groups that this group is a member of.

OWNER

Defines the user or group that owns the record.

PROFUSR

Displays a list of the users associated with this profile group.

PWD_AUTOGEN

Indicates whether the group password is automatically generated. The default is no. Used by CA SSO.

PWD_SYNC

Indicates whether the group password is automatically kept identical for all group applications. The default is no. Used by CA SSO.

PWPOLICY

Defines the record name of the password policy for the group. A password policy is a set of rules for checking the validity of a new password and for defining when a password expires. The default is no validity check. Used by CA SSO.

REVACL

Displays the accessor's access control lists.

SHELL

(UNIX only) The shell program assigned to a new UNIX user when the user is a member of this group.

Use the shellprog parameter with the chxgrp command to modify this property.

SUBGROUP

Displays the list of groups that have this group as a parent.

SUPGROUP

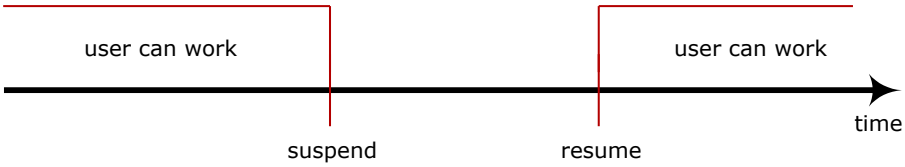
Defines the name of the parent group ("superior" group).

Use the parent[-] parameter with the ch[x]grp command to modify this property.

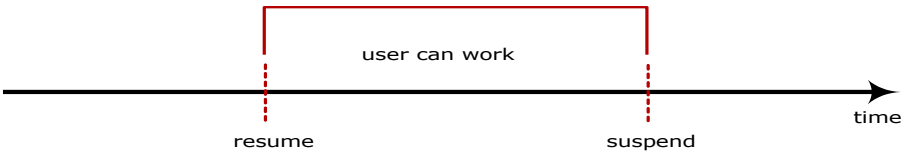
SUSPEND_DATE

Defines the date on which a user account is suspended and so becomes invalid.

If the suspend date for a record precedes its resume date, the user can work before the suspend date and after the resume date.



If a user has a resume date that is earlier than the suspend date, the record is also invalid *before* the resume date. The user can work only between the resume and suspend dates.



A value for the SUSPEND_DATE property in a user record overrides the value in a group record.

Note: This property corresponds to the suspend[-] parameter of the ch[x]usr and ch[x]grp commands.

SUSPEND_WHO

Displays the administrator who activated the suspend date.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

USERLIST

Displays the users that belong to the group.

The user list contained in this property may be different from the one in the native environment USERS property.

XUSER Class

Each record in the XUSER class defines an enterprise user in the database.

The key of the XUSER record is the name of the user—the name entered by the user when logging into the system.

You can change most of these properties from the CA Access Control Endpoint Management or by using the selang command chxusr.

Note: In most cases, and unless otherwise indicated, to change a property using chxusr, you use the property name as the command parameter.

You can view all properties from the CA Access Control Endpoint Management or by using the selang command showxusr.

APPLIST

Used by CA SSO.

APPLIST_TIME

Used by CA SSO.

APPLS

(Informational) Displays the list of applications that the accessor is authorized to access. Used by CA SSO.

AUDIT_MODE

Defines the activities that CA Access Control records in the audit log. You can specify any combination of the following activities:

- No logging
- All activities recorded in the trace file
- Unsuccessful login attempts
- Successful logins
- Failed access attempts to resources protected by CA Access Control
- Successful accesses to resources protected by CA Access Control

Note: This property corresponds to the audit parameter of the ch[x]usr and ch[x]grp commands.

AUTHNMTHD

(Informational) Displays the authentication method or methods to be used with the group record; from method 1 to method 32, or none. Used by CA SSO.

BADPASSWD

Used by CA SSO.

CALENDAR

Represents a Unicenter TNG calendar object for user, group, and resource restrictions in CA Access Control. CA Access Control fetches Unicenter TNG active calendars at specified time intervals.

CATEGORY

Defines one or more security categories assigned to a user or a resource.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

COUNTRY

A string that specifies a country descriptor for a user. This string is part of the X.500 naming scheme. CA Access Control does not use it for authorization.

CREATE_TIME

(Informational) Displays the date and time when the record was created.

DAYTIME

Defines the day and time restrictions that govern when an accessor can access a resource.

Use the restrictions parameter with the chres, ch[x]usr, or ch[x]grp commands to modify this property.

The resolution of daytime restrictions is one minute.

EMAIL

Defines the email address of the user, up to 128 characters.

FULLNAME

Defines the full name associated with an accessor. CA Access Control uses the full name to identify the accessor in audit log messages, but not for authorization.

FULLNAME is an alphanumeric string. For groups the maximum length is 255 characters. For users the maximum length is 47 characters.

GAPPLS

(Informational) Indicates the list of application groups that the user is authorized to access. Used by CA SSO.

GRACELOGIN

Defines the number of grace logins a user has after a password expires. When the number of grace logins is exceeded, the user is denied access to the system and must contact the system administrator for a new password.

The number of grace logins must be between 0 and 255. If this value is 0, the user cannot log in.

A value for the GRACELOGIN property in a USER record overrides a value for NGRACE in a GROUP record. Both override the PASSWDRULES property in the SEOS class record.

Note: This property corresponds to the grace parameter of the ch[x]usr command.

GROUPS

(Informational) Displays the list of user groups that the user belongs to. This property also contains any group authorities, such as group administration authority (GROUP-ADMIN), assigned to the user for each group the user belongs to.

The group list contained in this property may be different from the one in the native environment GROUPS property.

Note: This property is not modified by the ch[x]usr command. Instead, use the join[-] or joinx[-]command to modify this property.

INACTIVE

Defines the number of days of inactivity that must pass before the system changes the status of a user to inactive. If the account status is inactive, the user cannot log in.

A value for the INACTIVE property in a USER record overrides a value in a GROUP record. Both override the INACT property in the SEOS class record.

Note: CA Access Control does not store the status; it calculates the status dynamically. To identify inactive users, you must compare the INACTIVE value with the user's LAST_ACC_TIME value.

LAST_ACC_TERM

Displays the terminal from which the last login was performed.

LAST_ACC_TIME

Displays the date and time of the last login.

LOCALAPPS

Used by CA SSO.

LOCATION

Defines a user location. CA Access Control does not use this information for authorization.

LOGININFO

Defines the information needed to log the user into a specific application and audit data. LOGININFO contains a separate list for each application that the user is authorized to access. Used by CA SSO.

LOGSHIFT

Indicates whether a login outside of the shift time frame is permitted. CA Access Control writes an audit record in the audit log for this event.

MAXLOGINS

Defines the maximum number of concurrent logins that a user is allowed. A zero value indicates that the user can have any number of concurrent logins.

For a user to log in and administer the database (for example, by running selang), MAXLOGINS must be zero or greater than one. This is because CA Access Control considers each task (login, selang, GUI, and so forth) to be a terminal session.

A value for the MAXLOGINS property in a user record overrides a value in a group record. Both override the value of MAXLOGINS in the SEOS class record.

MIN_TIME

Defines the minimum time in days allowed between password changes for the user.

A value for the MIN_TIME property in a USER record overrides a value in a GROUP record. Both override the PASSWDRULES property in the SEOS class record.

Note: This property corresponds to the min_life parameter of the ch[x]usr command.

NOTIFY

Defines the user to be notified when a resource or user generates an audit event. CA Access Control can email the audit record to the specified user**.**

Limit: 30 characters.

OBJ_TYPE

Specifies the user authority attributes. Each of these attributes corresponds to the parameter of the same name in the ch[x]usr command. A user can have one or more of the following authority attributes:

ADMIN

Specifies whether the user can perform administrative functions, similar to root in the UNIX environment.

AUDITOR

Specifies whether the user can monitor the system, list information in the database, and set the audit mode for existing records.

IGN_HOL

Specifies whether the user can log in during any period of time defined in a HOLIDAY record.

LOGICAL

Specifies that the user is only for internal CA Access Control purposes and cannot be used by a real user to log in.

For example, the user nobody that you can use as the owner of resources to prevent even the resource owner from accessing the resource is a logical user by default. This means that no user can log in using this account.

OPERATOR

Specifies whether the user can list everything in the database and use the secons utility.

PWMANAGER

Specifies whether the user can modify the password settings of other users and can enable a user account that has been disabled by the serevu utility.

SERVER

Specifies whether a process can ask users for authorization and can issue the SEOSROUTE_VerifyCreate API call.

OIDCRDDATA

Used by CA SSO.

OLD_PASSWD

Contains an encrypted list of the user's previous passwords. The user cannot choose a new password from this list. The maximum number of passwords saved in OLD_PASSWD is determined by the setoptions command.

ORG_UNIT

A string that stores information on the organizational unit in which the user works. This string is part of the X.500 naming scheme. CA Access Control does not use it for authorization.

ORGANIZATION

Defines the organization in which the user works. This string is part of the X.500 naming scheme. CA Access Control does not use this for authorization.

PASSWD_A_C_W

Indicates the ADMIN user who last changed the user password for this record.

PASSWD_INT

Defines the maximum time in days between password changes for users.

A value for the PASSWD_INT property in a USER record overrides the value in a GROUP record. Both override the PASSWDRULES property in the SEOS class record.

Note: This property corresponds to the interval parameter of the ch[x]usr command.

PASSWD_L_A_C

Displays the date and time at which an administrator last updated the password.

PASSWD_L_C

Displays the date and time at which a user last updated the password.

PGMINFO

Defines the program information automatically generated by CA Access Control.

The Watchdog automatically verifies the information stored in this property. If it is changed, CA Access Control defines the program as untrusted.

You can select any of the following flags to *exclude* the associated information from this verification process:

crc

The cyclic redundancy check and MD5 signature.

ctime

(UNIX only) The time of the last file status change.

device

On UNIX, the logical disk that the file resides on. On Windows, the drive number of the disk containing the file.

group

The group that owns the program file.

inode

On UNIX, the file system address of the program file. On Windows, this has no meaning

mode

The associated security protection mode for the program file.

mtime

The time the program file was last modified.

owner

The user who owns the program file.

sha1

The SHA1 signature. Digital signature method called Secure Hash Algorithm that could be applied to the program or sensitive files.

size

The size of the program file.

Use the flags, flags+, or flags- parameter with the chres, editres, or newres command to modify the flags in this property.

PHONE

Defines the user's telephone number. This information is not used for authorization.

PWD_AUTOGEN

Displays whether the user password is automatically generated. Used by CA SSO.

The default is no.

PWD_SYNC

Displays whether the user password is automatically kept identical for all user applications. Used by CA SSO.

The default is no.

RESUME_DATE

Defines the date on which a suspended USER account becomes unsuspended.

RESUME_DATE and SUSPEND_DATE work together.

Note: This property corresponds to the resume[-] parameter of the ch[x]usr and ch[x]grp commands.

REVACL

Displays the access control lists of the accessor.

REVOKE_COUNT

Used by CA SSO.

SCRIPT_VARS

Used by CA SSO, Defines a variables list with the variable values of the application script that are saved per application.

SECLABEL

Defines the security label of a user or resource.

Note: The SECLABEL property corresponds to the label[-] parameter of the chres and ch[x]usr commands.

SECLEVEL

Defines the security level of an accessor or resource.

Note: This property corresponds to the level[-] parameter of the ch[x]usr and chres commands.

SESSION_GROUP

Defines an SSO session group for a user. The SESSION_GROUP property is a string with a maximum length of 16 characters.

In Windows, an administrator can enter a session group new name if the preferred name is not in the drop-down list.

Used by CA SSO.

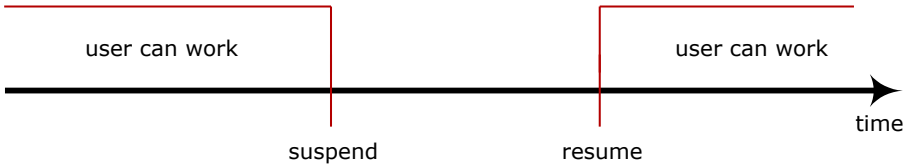
SHIFT

Used by CA SSO.

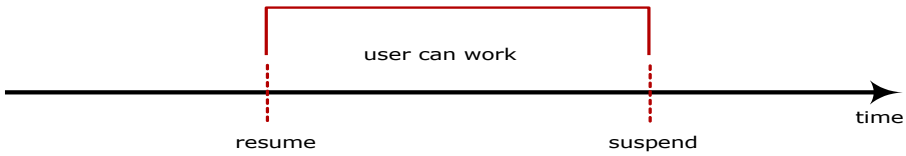
SUSPEND_DATE

Defines the date on which a user account is suspended and so becomes invalid.

If the suspend date for a record precedes its resume date, the user can work before the suspend date and after the resume date.



If a user has a resume date that is earlier than the suspend date, the record is also invalid *before* the resume date. The user can work only between the resume and suspend dates.



A value for the SUSPEND_DATE property in a user record overrides the value in a group record.

Note: This property corresponds to the suspend[-] parameter of the ch[x]usr and ch[x]grp commands.

SUSPEND_WHO

Displays the administrator who activated the suspend date.

UALIAS

Displays the aliases of a specific user defined to one or more authentication hosts. Used by CA SSO.

UPDATE_TIME

(Informational) Displays the date and time when the record was last modified.

UPDATE_WHO

(Informational) Displays the administrator who performed the update.

Classes in the Windows Environment

This section contains a complete alphabetic reference to all the Windows classes and properties that exist in the Windows database (classes in the nt environment).

Note: The term *nt environment* refers to the database accessed with the selang command env nt. This is the same database the Windows operating system maintains for users, groups, and resources.

COM Class

Each record in the COM class defines a device specifying a serial port (COM) or a parallel port (LPT) as listed in the Windows Control Panel, Ports.

Note: You cannot create new objects in the COM class using CA Access Control.

The key of the COM class is the name of the port being controlled.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

DEV

(Informational). A string to indicate the device serial number.

DACL

Defines the standard access control list that contains the user names and group names authorized to access the resource, and the level of access granted to each.

Users who want to modify this property must be the owner of the resource or have special access to the resource (to modify the ACL).

Each element in the access control list contains the following information:

Access Type

Specifies permissions to the resource:

- **Allowed**-Permits special access to the resource.
- **Denied**-Denies special access to the resource.

Accessor

The user or group for whom the access rights are allowed or denied.

Access

The access authority that the accessor has to the resource.

Note: In an empty ACL, no accesses are explicitly granted, so access is implicitly denied. For a resource that has no ACL, no protection is assigned to the object, so any access request is granted.

Use auth or auth- command to modify this property.

GID

Displays the group information for the file or device.

OWNER

Defines the user or group that owns the record.

SACL

Windows System Access Control List. Displays audit directives.

DEVICE Class

Each record in the DEVICE class defines a Windows hardware device as listed in the Windows Control Panel, Devices.

The key of the DEVICE class record is the name of the device being controlled.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

STARTUPTYPE

Defines how (when) the device is started. Options are:

automatic

Starts the device automatically during system startup.

boot

Starts the device every time the system starts, before any other devices start. Select this option for critical devices essential to system operation.

disabled

Prevents users from starting the device. The system can still start disabled devices.

manual

Allows the device to be started by a user or a dependent device.

system

Starts the device every time the system starts, after the Boot devices start. Select this option for critical devices essential to system operation.

Use the starttype parameter with the chres or editres commands to modify this property.

STATUS

Changes the current service state. Options are: started, stopped, and paused.

Use the status parameter with the chres or editres commands to modify this property.

IMAGEPATH

The fully qualified path for the specified device.

PROFILE

A string that specifies a path to the user's profile. This string can include a local absolute path, or a UNC path.

Use the profile parameter with the chusr, editusr, or newusr command to modify this property.

Example: Activate a modem

To display the status of the modem, enter the selang command:

```
showres DEVICE modem
```

To activate the modem, enter the command:

```
chres device modem status(started)
```

DISK Class

Each record in the DISK class defines a system volume. Volume is the general term that refers to any of the entities that you can create and use on a computer running Windows operating systems (Server editions) such as a primary partition, a logical drive in an extended partition, a volume set, a stripe set, a mirror set, or a stripe set with parity. A volume has a single drive letter assigned to it and is formatted for use by a file system.

Note: You cannot create objects in the DISK class using CA Access Control.

The key of the DISK class is the assigned drive letter (C:, D:, an so on).

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

ATIME

(Informational). The time the record was last accessed.

CTIME

(Informational). Created time.

DACL

Defines the standard access control list that contains the user names and group names authorized to access the resource, and the level of access granted to each.

Users who want to modify this property must be the owner of the resource or have special access to the resource (to modify the ACL).

Each element in the access control list contains the following information:

Access Type

Specifies permissions to the resource:

- **Allowed**-Permits special access to the resource.
- **Denied**-Denies special access to the resource.

Accessor

The user or group for whom the access rights are allowed or denied.

Access

The access authority that the accessor has to the resource.

Note: In an empty ACL, no accesses are explicitly granted, so access is implicitly denied. For a resource that has no ACL, no protection is assigned to the object, so any access request is granted.

Use auth or auth- command to modify this property.

FILE_SYSTEM

(Informational). A name to designate the file system (such as FAT or NTFS).

FREE_SPACE

(Informational). The total amount of free space (in KB) on the disk.

GID

Displays the group information for the file or device.

LABEL

(Informational). The name of the specified volume.

LINK_NUMB

(Informational). Specifies the number of links. For non-NTFS file systems, this property is always one.

MTIME

(Informational). The time the record was last modified.

OWNER

Defines the user or group that owns the record.

SACL

Windows System Access Control List. Displays audit directives.

TYPE

(Informational). Specifies whether the disk is removable, fixed, a CD-ROM, a RAM disk, or a network drive.

USED_SPACE

(Informational). The total amount of used space (in KB) on the disk.

DOMAIN Class

Each record in the DOMAIN class defines a collection of computers that share a common database and security policy (domain). A domain provides access to the centralized user accounts and group accounts maintained by the domain administrator. Each domain has a unique name.

Note: You cannot create new objects in the DOMAIN class using CA Access Control.

The key to the DOMAIN record is the domain name.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

BDC

(Informational). The name of the computer that receives a copy of the domain's directory database and contains all account and security policy information for the domain. The copy is synchronized periodically and automatically with the master copy on the primary domain controller (PDC). Backup domain controllers (BDCs) also authenticate user logins and can be promoted to function as PDCs as needed. Multiple BDCs can exist on a domain.

COMPUTERS

Lists computers that are the members of the specified domain.

Use computer or computer- parameter with the chres and editres commands to modify this property.

DOMAIN_NAME

Defines the domain name.

DOMAIN_USERS

(Informational). Lists user and group accounts that are members of the specified domain.

PDC

(Informational). The name of the first computer created in the domain; this computer contains the primary storehouse for domain data. It authenticates domain logins and maintains the directory database for a domain. The primary domain controller (PDC) tracks changes made to accounts of all computers on a domain. It is the only computer to receive these changes directly. A domain has only one PDC.

TRUSTED

Lists trusted and trusting domains.

A trust relationship is a link between domains that allows pass-through authentication, in which a trusting domain honors the login authentications of a trusted domain. With trust relationships, a user with only one user account in one domain can potentially access the entire network. You can give user accounts and global groups defined in a trusted domain rights and resource permissions in a trusting domain, even though those accounts do not exist in the trusting domain's directory database.

Use the trusted or trusting- parameter with the chres and editres commands to modify this property. You should specify a password for this command.

TRUSTING

The Trusting domain are domains which trust the target domain.

FILE Class

Valid in the Windows environment

Each record in the FILE class defines a file on a file system (for example, FAT, NTFS, or CDFS) on a physical or logical drive of a computer.

Note: You cannot use CA Access Control to physically create files on disk.

The key of the FILE class record is the name of the file or directory protected by the record. The full path must be specified.

The following definitions describe the properties contained in a FILE record. You can use selang or the Web based GUI to change the record's modifiable properties.

ATIME

Displays the time the file was last accessed.

ATTRIB

Displays attributes for the file or directory. The attributes can be one or more of the following:

- ARCHIVE
- COMPRESSED
- DIRECTORY
- HIDDEN
- NORMAL
- OFFLINE
- READONLY
- SYSTEM
- TEMPORARY

CTIME

Displays the created time.

DACL

Defines the standard access control list that contains the user names and group names authorized to access the resource, and the level of access granted to each.

Users who want to modify this property must be the owner of the resource or have special access to the resource (to modify the ACL).

Each element in the access control list contains the following information:

Access Type

Specifies permissions to the resource:

- **Allowed**-Permits special access to the resource.
- **Denied**-Denies special access to the resource.

Accessor

The user or group for whom the access rights are allowed or denied.

Access

The access authority that the accessor has to the resource.

Note: In an empty ACL, no accesses are explicitly granted, so access is implicitly denied. For a resource that has no ACL, no protection is assigned to the object, so any access request is granted.

Use auth or auth- command to modify this property.

DEV

Displays the serial number of the volume where the file is located.

FILE_SYSTEM

Displays the name of the file system where the file is located.

GID

Displays the group information for the file or device.

INDEX

Displays the unique identifier associated with the file.

ISDIR

Indicates whether the file is a directory.

LINKS_NUMB

Displays the number of links to the file. For the FAT file systems, this property is always one. For NTFS, it can be more than one.

MTIME

Displays the time the file was last modified.

NAME

Displays the file name.

OWNER

Defines the user or group that owns the record.

SACL

Windows System Access Control List. Displays audit directives.

SIZE

Displays the size of the file in bytes.

More information:

Windows File Attributes (see page 475)
chfile Command—Modify Windows File Settings (see page 180)

GROUP Class

The GROUP class contains all group records defined to the Windows operating system. A record in the GROUP class represents every group of users.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked as *informational* and cannot be modified.

COMMENT

Additional information you want to include in the record. CA Access Control does not use this information for authorization.

Use the comment[-] parameter with the chgrp, editgrp, and newgrp commands to modify this property.

Limit: 255 characters.

FULL_NAME

The full name associated with a user. CA Access Control uses the full name to identify the user in audit log messages, but not for authorization.

Use the name parameter with the chusr, editusr, or newusr command to modify this property.

GID

(Informational). A value that contains the relative identifier of the group. The relative identifier is determined by the accounts database when the group is created. It uniquely identifies the group to the account manager within the domain.

GLOBAL

Indicates a global group. This property is only applicable to Windows groups. It replaces the ISGLOBAL property of earlier CA Access Control versions.

Use the global parameter with the newgrp (only) command to add this property.

USERLIST

The list of users and global groups (for local groups only) that belong to the group. The list contained in this property may be different from the one in the CA Access Control database.

Use the username(*groupname)* parameter with the join[-] command to modify this property.

PRIVILEGES

The Windows rights assigned to the group.

Use the privileges parameter with the chgrp, editgrp, or newgrp command to modify this property.

More information:

chgrp Command—Modify Windows Groups (see page 181)
Windows Privileges (see page 478)

OU Class

The OU (Organizational Unit) class contains objects such as user, group, or computer. Objects of class OU can be created on the primary domain controller and could have other objects as child objects (such as group), so an object of class OU is a container object.

Note: The OU class is available only for Windows 2000 Advanced Server with Active Directory installed.

The OU class has no predefined properties (like other classes have). However, you can update the following OU properties:

- Country/Region
- Description
- Desktop
- City
- Display Name
- Folder (Read-only property)
- Fax number
- Managed objects (Read-only property)
- Member of (Read-only property)
- Name (Read-only property)
- Postal address
- Postal code
- P.O. box
- State/Province
- Street
- Telephone
- Object changed (Read-only property)
- Object created (Read-only property)
- Web page

PRINTER Class

Each record in the PRINTER class defines a device connected to a Windows computer system that is capable of reproducing a visual image on a medium (as listed in Printers folder)

Note: You cannot create new objects of class PRINTER using CA Access Control.

The key of the PRINTER class record is the name of the local printer.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

DACL

Defines the standard access control list that contains the user names and group names authorized to access the resource, and the level of access granted to each.

Users who want to modify this property must be the owner of the resource or have special access to the resource (to modify the ACL).

Each element in the access control list contains the following information:

Access Type

Specifies permissions to the resource:

- **Allowed**-Permits special access to the resource.
- **Denied**-Denies special access to the resource.

Accessor

The user or group for whom the access rights are allowed or denied.

Access

The access authority that the accessor has to the resource.

Note: In an empty ACL, no accesses are explicitly granted, so access is implicitly denied. For a resource that has no ACL, no protection is assigned to the object, so any access request is granted.

Use auth or auth- command to modify this property.

COMMENT

Defines additional information that you want to include in the record. CA Access Control does not use this information for authorization.

Limit: 255 characters.

LOCATION

A string that indicates the printer location. CA Access Control does not use this information for authorization.

Use the location parameter with the chres or editres commands to modify this property. Use () with blanks to delete this property.

OWNER

Defines the user or group that owns the record.

SHARE

The name that identifies the share point for the printer. Users or groups that want to access the printer could use its share name.

Use the share_name or share_name- parameter with the chres or editres commands to modify this property.

NAME

Printer name.

SACL

Windows System Access Control List. Displays audit directives.

SERVER

(Informational). A string to identify the server that controls the printer. If there is no such property, the printer is controlled locally.

PROCESS Class

Each record in the PROCESS class defines an object consisting of an executable program, a set of virtual memory addresses, and a thread as listed in the Windows Task Manager.

Note: You cannot create new objects in the PROCESS class using CA Access Control.

The key of the PROCESS class record is the name of the executable module of the running program.

The following definitions describe the properties contained in this class record. There are no modifiable properties in this class. Non-modifiable properties are marked *informational*.

IMAGE_PATH

(Informational). The fully qualified path for the specified executable module.

PROCESS_ID

(Informational). The unique identifier of the process. Process ID numbers are reused, so they identify a process only for the lifetime of that process.

REGKEY Class

Each record in the REGKEY class defines a key in the Windows registry.

The key to the REGKEY record is the full registry path to the Windows registry key.

Note: You can use wildcard characters as part of the path specification.

The following definitions describe the properties contained in a REGKEY record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

DACL

The standard access control list that contains the user names and group names authorized to access the resource and the level of access granted to each.

Users who want to modify this property must be the owner of the resource or have special access to the resource (to modify the ACL).

Each element in the access control list contains the following information:

Access Type

Specifies permissions to the resource:

- **Allowed-**Permits special access to the resource
- **Denied-**Denies special access to the resource

Accessor

The name of the user or group for whom the access rights are allowed or denied.

Access

The access authority the accessor has to the resource. Valid access authorities for the REGKEY class are:

- **all**-Allows or denies the accessor to perform all operations permissible for the class
- **append**/**create**/**subkey**-Allows or denies the accessor to create or modify a subkey of the registry key
- **changeperm**/**sec**/**dac**/**writedac**/**perm**-Allows or denies the accessor to modify the ACL (that is, add or remove accessors) of a resource.
- **chown**/**owner**/**takeownership**-Allows or denies the accessor to change the owner of the resource
- **delete**-Allows or denies the accessor to delete a resource
- **enum**-Allows or denies the accessor to enumerate subkeys of the registry key
- **link**-Allows or denies the accessor to create link to a registry key
- **notify**-Allows or denies the accessor to request change notifications for a registry key or for subkeys of a registry key
- **query**-Allows or denies the accessor to query a value of the registry key
- **read**-Allows or denies the accessor to read the key's contents, but prevents changes from being saved
- **readcontrol/manage**-Allows or denies the accessor to read the information in the registry key's security descriptor, not including the information in the system (audit) ACL
- **set**-Allows or denies the accessor to create or set a value of the registry key
- **write**-Allows or denies the accessor to change the registry key and its subkeys

Note: It is important to note the differences between an ACL that is empty (that is, one that has no entries) and a resource without an ACL. In the case of an empty ACL, no accesses are explicitly granted, so access is implicitly denied. For a resource that has no ACL, no protection is assigned to the object, so any access request is granted.

Use auth or auth- command to modify this property.

OWNER

The user or group designated as the owner of the resource.

Use the owner parameter with the newres, chres, and editres commands to modify this property.

SACL

Windows System Access Control List specifies audit directives.

SUBKEYS

(Informational). A list of registry keys (subkeys) located under the key.

SUBVALUES

(Informational). A list of registry values described in the current registry key.

REGVAL Class

Each record in the REGVAL class defines data that describes the registry keys. This data stores information necessary to configure the system for one or more users, applications, and hardware devices. Registry values contain information that is constantly referenced during operation. Examples include:

- Profiles for each user
- Applications installed on the computer and the types of files each can create
- Property sheet settings for folders and application icons
- Hardware configuration
- Used ports

The key to the REGVAL record is the full registry key name and its value.

Note: Changing or deleting registry keys and their values incorrectly can cause serious, system-wide problems that may require you to reinstall Windows to correct them.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

TYPE

A format to store data. When you store data under a registry value you can specify one of the following values to indicate the type of data being stored:

Note: Specify the type when you create or modify the registry value.

DWORD

Data represented by a number that is four bytes long. Many parameters for device driver and services are this type, and can be displayed in binary, hexadecimal, or decimal format.

STRING

A sequence of characters representing readable text

MULTISTRING

A multiple string. Values that contain lists or multiple values in readable text. Entries are separated by null characters.

BINARY

Raw, binary data. Most hardware component information is stored as binary data and can be displayed in hexadecimal format or in an easy-to-read format.

Use one of these described types as a parameter with the newres, chres or editres to modify this property.

VALUE

The value that the Windows registry value holds.

SEOS Class

The SEOS class controls the behavior of the native local security system.

The class contains only one record, called SEOS, which specifies general native security options. To view or change the status of SEOS class properties, use the setoptions command.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

AuditCategory

Specifies which detected authorized and unauthorized events are audited.

AccountLogon

Specifies whether to audit each instance of a user logging on to or logging off from another computer in which this computer is used to validate the account.

AccountManagement

Specifies whether to audit each event of account management on a computer. Examples of account management events include:

- A user account or group is created, changed, or deleted.
- A user account is renamed, disabled, or enabled.
- A password is set or changed.

DirectoryAccess

Specifies whether to audit the event of a user accessing an Active Directory object that has its own system access control list (SACL) defined.

Logon

Specifies whether to audit each instance of a user logging on to or logging off from a computer.

ObjectAccess

Specifies whether to audit the event of a user accessing an object. For example, a file, folder, registry key, printer, and so on, that has its own system access control list (SACL) defined.

PolicyChange

Specifies whether to audit every incident of a change to user rights assignment policies, audit policies, or trust policies.

PrivilegeUse

Specifies whether to audit each instance of a user exercising a user right.

DetailedTracking

Specifies whether to audit detailed tracking information for events such as program activation, process exit, handle duplication, and indirect object access.

System

Specifies whether to audit when a user restarts or shuts down the computer or when an event occurs that affects either the system security or the security log.

History

Defines the number of unique new passwords that have to be associated with a user account before an old password can be reused.

Limits: An integer between 1 and 24. If you specify zero, no passwords are saved.

Interval

Defines the period of time (in days) that a password can be used before the system requires the user to change it.

Min life

Defines the period of time (in days) that a password must be used before the user can change it.

Min length

Defines the least number of characters that a password for a user account may contain.

Password fails

Defines the number of failed logon attempts that causes a user account to be locked out.

Reset count after

Defines the number of minutes that must elapse after a failed logon attempt before the failed logon attempt counter is reset to 0 bad logon attempts.

SERVICE Class

Each record in the SERVICE class defines a Windows service as listed in the Windows Control Panel, Services.

The key of the SERVICE class record is the name of the service being controlled.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

ACCOUNT

Changes the login account for the service. Although most services must log in to the system account, some services can be configured to log in to special user accounts. For more information, see the relevant Microsoft Windows documentation. The default value is LocalSystem.

Use the account parameter with the chres or editres commands to modify this property.

BINARY_NAME

The full path which points to the location of the service's executable.

IMAGEPATH

The fully qualified path for the specified executable module.

INTERACTIVE

Provides a user interface on the desktop that can be used by whoever is logged in when the service is started. This is available only if the service is running as a LocalSystem account.

Use the interactive parameter with the chres or editres commands to modify this property.

PROFILE

A string that specifies the path to the user's profile. This string can include a local absolute path, or a UNC path.

Use the profile parameter with the chusr, editusr, or newusr command to modify this property.

REG_KEY

This property points to the location of the service definition in Windows registry.

STARTUPTYPE

Defines how (when) the service is started. Options are:

- **automatic**-Starts automatically during system startup.
- **disabled**-Prevents users or dependent services from starting the service.
- **manual**-Allows the service to be started by a user or a dependent service.
- Use the startuptype parameter with the chres or editres commands to modify this property.

STATUS

Changes the current service state. Options are: started, stopped, and paused.

Use the status parameter with the chres or editres commands to modify this property.

Example: Configure a service to start manually

To change the service SeOSAgent to start manually, enter the selang command:

```
chres SERVICE "SeosAgent" starttype(manual)
```

Example: Change a directory login account

To change the login account of the Directory Replicator to ReplAdmin with password abcde, enter the selang command:

```
chres SERVICE directory replicator account(repladmin) domainpwd(abcde)
```

SESSION Class

Each record in the SESSION class defines a user session on the local host. The record includes the user name, computer name, elapsed time of the connection, and the resources being used.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

CNAME

The host name where the session was established.

GUEST

Indicates whether the session was created on Guest account.

IDLE

Ends a network session between a server and a workstation.

Use the disconnect parameter with the chres or editres commands to modify this property.

OPENS

Indicate the number of open sessions.

RESOURCES

A property that gives information about shared files on a server. This information includes the path of the opened shared resource and the user or computer that opened the resource.

TIME

The time elapsed since the session was established.

USER

A value that contains the relative ID (RID) of the user. The RID is determined by the Security Account Manager (SAM) when the user is created. It uniquely defines the user account to SAM within the domain.

Example: Disconnect a user from a local session

To disconnect user ZORRO from a session on the local host, enter the selang command:

```
chres SESSION zorro disconnect
```

Note: Disconnecting users may result in loss of data. It is a good idea to warn connected users before disconnecting them.

SHARE Class

Each record in the SHARE class defines a share resource that could be any device, data, or program used by one or more devices or programs. For Windows, shared resources refer to any resource that is made available to network users, such as directories, files, printers, and named pipes. A share also refers to a resource on a server that is available to network users.

The key of the SHARE class record is the share name of the resource.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

CURR_USERS

(Informational). The number of current connections to the resource.

DACL

Defines the standard access control list that contains the user names and group names authorized to access the resource, and the level of access granted to each.

Users who want to modify this property must be the owner of the resource or have special access to the resource (to modify the ACL).

Each element in the access control list contains the following information:

Access Type

Specifies permissions to the resource:

- **Allowed**-Permits special access to the resource.
- **Denied**-Denies special access to the resource.

Accessor

The user or group for whom the access rights are allowed or denied.

Access

The access authority that the accessor has to the resource.

Note: In an empty ACL, no accesses are explicitly granted, so access is implicitly denied. For a resource that has no ACL, no protection is assigned to the object, so any access request is granted.

Use auth or auth- command to modify this property.

MAX_USERS

The maximum number of concurrent connections that the shared resource can accommodate.

Note: You cannot supply zero (0) as a value for this property. Windows ignores it.

Use the max_users parameter with the newres, chres, or editres commands to modify this property.

NAME

Defines the name of the share.

PATH

A string that specifies a local path for the shared resource. For disks, this is the path being shared. For print queues, this is the name of the print queue being shared.

Use the path parameter with the newres, chres, or editres commands to modify this property.

PERMISSION

(Informational). A value that indicates the shared resource's permissions for servers running with share-level security. This property can be any of the values in the following table:

ACCESS_READ

Permission to read data from a resource and, by default, to execute the resource.

ACCESS_WRITE

Permission to write data to the resource.

ACCESS_CREATE

Permission to create an instance of the resource (such as a file); data can be written to the resource as the resource is created.

ACCESS_EXEC

Permission to execute the resource.

ACCESS_DELETE

Permission to delete the resource.

ACCESS_ATRIB

Permission to modify the resource's attributes (such as the date and time when a file was last modified).

ACCESS_PERM

Permission to modify the permissions (read, write, create, execute, and delete) assigned to a resource for a user or application.

ACCESS_ALL

Permission to read, write, create, execute, and delete resources, and to modify their attributes and permissions.

ACCESS_NONE

Denies permissions.

REMARK

Additional information you want to include in the record. The alphanumeric string can contain up to 255 characters. CA Access Control does not use this information for authorization.

Use the comment or comment- parameter with the newres, chres, or editres commands to modify this property.

RESOURCES

(Informational). A property that gives information about shared files on a server. This information includes the path of the opened shared resource and the user or computer that opened the resource.

TYPE

(Informational). The type of share. Use one of the following types for a shared resource:

File Folder

A disk drive. This can also refer to remote administration of the server (ADMIN$) and to administrative shares such as C$, D$, and so on.

Print Queue

A print queue

Communication device

A communication device

Interprocess Communication (IPC)

A special share reserved for interprocess communication (IPC$)

USERS

Information about users currently accessing the shared resource. This information includes the name of user who made the connection (USER), the share name of the server's shared resource, or the computer name of the client (MACHINE). It also includes the number of seconds that the connection has been established (TIME) and the number of files currently open as a result of the connection (INUSE).

USER Class

The USER class contains all user records defined to the Windows operating system. The key of the USER record is the user's name, which is the name the user entered when logging into the system.

The following definitions describe the properties contained in this class record. Most properties are modifiable and can be manipulated using selang or the administration interfaces. Non-modifiable properties are marked *informational*.

BAD_PW_COUNT

(Informational). The number of times the user tried to log in to the account using an incorrect password. A value of -1 indicates that the value is unknown.

COMMENT

Additional information you want to include in the record. CA Access Control does not use this information for authorization.

Use the comment[-] parameter with the chusr, editusr, and newusr commands to modify this property.

Limit: 255 characters.

COUNTRY

A string that specifies a country descriptor for a user. This string is part of the X.500 naming scheme. CA Access Control does not use it for authorization.

Use the country parameter with the chusr, editusr, and newusr commands to modify this property.

DAYTIME

The day and time restrictions that govern when a user can access the resource.

Use the restrictions parameter with the chusr, editusr, and newusr commands to modify this property.

Note: The information in this property is identical to that in the DAYTIME property in the AC environment, except that any minute value entered is truncated.

DIAL_CALLBACK

The type of call-back privileges provided to the user. The following options are defined:

NoCallBack

The user has no call-back privileges.

SetByCaller

The remote user can specify a call-back phone number when dialing in.

Call-back Phone Number

The administrator sets the call-back number.

Use the gen_prop or gen_val parameters with the chusr or editusr command to modify this property.

DIAL_PERMISSION

Permission to dial in to the RAS server. When you specify 0 as value, the user cannot dial in to the RAS server.

Use the gen_prop or gen_val parameter with the chusr or editusr command to modify this property.

EXPIRE_DATE

The date on which a USER record expires and becomes invalid. A value for the EXPIRE_DATE property in a USER record overrides a value in a GROUP record. To reinstate the expired record, use the chusr command with the expire- parameter. You cannot resume an expired user. You can resume a suspended user by specifying a resume date.

Use the expire or expire- parameter with the chusr, editusr, or newusr command to modify this property.

FLAGS

Flags that you can assign to a user's account to specify particular attributes. You can apply more than one flag to each account.

Use the flags parameter with the chusr, editusr, and newusr commands to modify this property.

FULL_NAME

The full name associated with a user. CA Access Control uses the full name to identify the user in audit log messages, but not for authorization.

Use the name parameter with the chusr, editusr, or newusr command to modify this property.

GID

A value that contains the relative identifier of the group. The relative identifier is determined by the accounts database when the group is created. It uniquely identifies the group to the account manager within the domain.

GROUPS

The list of groups a user belongs to. The group list contained in this property may be different from the one in the AC environment GROUPS property.

Use the group parameter with the join[-] command to modify this property.

HOME

The home directory is the folder that is accessible to the user and contains files and programs for that user. The home directory can be assigned to individual user or shared among many users.

HOMEDIR

A string specifying the user's home directory. Users log in to their home directories automatically.

Use the homedir parameter with the chusr, editusr, or newusr command to modify this property.

HOME_DRIVE

A string that specifies the drive of the user's home directory. Users log in to their own home drives and home directories automatically.

Use the homedrive parameter with the chusr, editusr, or newusr command to modify this property.

ID

A value that contains the relative ID (RID) of the user. The RID is determined by the Security Account Manager (SAM) when the user is created. It uniquely defines the user account to SAM within the domain.

LAST_ACC_TIME

(Informational). The date and time of the last login.

LAST_LOGOFF

(Informational). The date and time of the last logoff.

LOCATION

A string used to store a user location. CA Access Control does not use this information for authorization.

Use the location parameter with the chusr, editusr, and newusr commands to modify this property.

LOGON_SERVER

A string that specifies the server that verifies the login information for the user. When the user logs into the domain workstation, CA Access Control transfers the login information to the server, which gives the workstation permission for the user to work.

MAX_LOGINS

(Informational). The number of times the user logged in successfully to this account. A value of -1 indicates that the value is unknown.

NAME

The name of the user.

ORGANIZATION

A string that stores information on the organization in which the user works. This string is part of the X.500 naming scheme. CA Access Control does not use it for authorization.

Use the organization parameter with the chusr, editusr, and newusr commands to modify this property.

ORG_UNIT

A string that stores information on the organizational unit in which the user works. This string is part of the X.500 naming scheme. CA Access Control does not use it for authorization.

Use the org_unit parameter with the chusr, editusr, and newusr commands to modify this property.

PASSWD_EXPIRED

Expiration date for the user account.

PGROUP

A user's primary group ID. A primary group is one of the groups in which a user is defined. A primary group must be a global group. This string cannot include spaces or commas.

Use the pgroup parameter with the chusr, editusr, or newusr command to modify this property.

PHONE

A string that can be used to store a user telephone number. This information is not used for authorization.

Use the phone parameter with the chusr, editusr, and newusr commands to modify this property.

PRIVILEGES

The Windows rights assigned to the user.

Use the privileges parameter with the chusr, editusr, or newusr command to modify this property.

PROFILE

A string that specifies a path to the user's profile. This string can include a local absolute path, or a UNC path.

Use the profile parameter with the chusr, editusr, or newusr command to modify this property.

PW_LAST_CHANGE

(Informational). The date and time on which the password was updated.

RESUME_DATE

The date on which a suspended USER account becomes valid.

See SUSPEND_DATE for an explanation of how RESUME_DATE and SUSPEND_DATE work together.

SCRIPT

A string that specifies the path for the user's logon script file. The script file can be a .CMD, .EXE , or .BAT file.

TERMINALS

A string that specifies a list of terminals from which the user can log in.

Use the terminals parameter with the chusr, editusr, and newusr commands to modify this property.

TS_CONFIG_PGM

A value that indicates whether the client can specify the initial program.

The TS_INITIAL_PGM user property indicates the initial program. If you specify a user's initial program, it becomes the only program that user can run; terminal server logs off the user when the user exits that program.

When this value is set to 1, the client can specify the initial program. When this value is set to 0, the client cannot specify the initial program.

Use the gen_prop and gen_val parameters with the chusr and editusr commands to modify this property.

TS_HOME_DIR

The path of the user's home directory for terminal server logon. This string can specify a local path or a UNC path (\\machine\share\path).

Use the gen_prop and gen_val parameters with the chusr and editusr commands to modify this property.

TS_HOME_DRIVE

A drive specification (a drive letter followed by a colon) to which the UNC path is specified in the TS_HOME_DIR property.

Use the gen_prop and gen_val parameters with the chusr and editusr commands to modify this property.

TS_INITIAL_PGM

The path of the initial program that Terminal Services runs when the user logs on.

If you specify a user's initial program, that is the only program that user can run. Terminal server logs off the user when the user exits that program.

When TS_CONFIG_PGM property is set to 1, the client can specify the initial program.

Use the gen_prop and gen_val parameters with the chusr and editusr commands to modify this property.

TS_PROFILE_PATH

The path of the user's profile for terminal server logon. The directory identified by the path must be created manually and must exist prior to the logon.

Use the gen_prop and gen_val parameters with the chusr and editusr commands to modify this property.

TS_WORKING_DIR

The path of the working directory for the initial program that Terminal Services runs when the user logs on.

Use the gen_prop and gen_val parameters with the chusr and editusr commands to modify this property.

WORKSTATIONS

A list of the workstations from which the user can log in.

Use the workstations parameter with the chusr, editusr, and newusr commands to modify this property.

More information:

chusr Command—Modify Windows Users (see page 186)
Windows Account Flags (see page 476)
Windows Privileges (see page 478)

Classes in the UNIX Environment

This section contains a complete alphabetic reference to all the UNIX classes that exist in the UNIX system files (classes in the unix environment). The properties for these native classes are governed by the operating system and vary between systems.

Note: The term *unix environment* refers to the system files accessed with the selang command env unix. These are the same system files the UNIX operating system maintains for users and groups and the files on the system.

FILE Class

Each record in the FILE class defines a file located on a physical or logical drive of a computer on a file system.

Note: You cannot create files physically on disk using CA Access Control.

The key of the FILE class record is the name of the file or directory protected by the record. The full path must be specified.

The properties for the this native class are governed by the operating system and vary between systems. The chfile command lists the native properties you can modify using selang.

GROUP Class

The GROUP class contains all group records defined to the UNIX operating system. A record in the GROUP class represents every group of users.

The properties for the this native class are governed by the operating system and vary between systems. The chgrp command lists the native properties you can modify using selang.

USER Class

The USER class contains all user records defined to the UNIX operating system. The key of the USER record is the user's name, which is the name entered by the user when logging into the system.

The properties for the this native class are governed by the operating system and vary between systems. The chusr command lists the native properties you can modify using selang.

Classes for Custom Purposes

This section contains a reference user defined classes and properties.

User Defined Class

Each record in the User Defined class defines access to a custom-made class that meets your own needs. The only restriction on the name of user-defined classes is that the name cannot be all uppercase letters.

The key of a User Defined class record is the name of the record.

Unicenter TNG User-Defined Class

CA Access Control lets you define Unicenter TNG asset classes as resources. You can create, delete, activate, and disable the Unicenter TNG user-defined classes.

You can find Unicenter TNG user-defined classes in the UACC class.

Note: Any property defined for a regular CA Access Control class can be used in User Defined classes.

Appendix A: Windows Values

This section contains the following topics:

Windows File Attributes (see page 475)
Windows Account Flags (see page 476)
Windows Permissions (see page 477)
Windows Privileges (see page 478)

Windows File Attributes

Attributes can be assigned to a file by using the chfile, editfile, and newfile commands. Attributes determine the character of the file.

Note: Although the full name for these file attributes is FILE_ATTRIBUTE_*name*, CA Access Control only requires you to enter the *name* portion (for example, ARCHIVE or COMPRESSED).

The following lists and describes the file attributes that you can modify in Windows.

FILE_ATTRIBUTE_ARCHIVE

An archival file; a file marked for backup or removal.

FILE_ATTRIBUTE_HIDDEN

A hidden file. Hidden files are not normally included in an ordinary directory listing.

FILE_ATTRIBUTE_NORMAL

A file with no other attributes. This value is only valid when used alone.

FILE_ATTRIBUTE_READONLY

A read-only file. Applications can read the file, but cannot write in it or delete it.

FILE_ATTRIBUTE_SYSTEM

An operating system file or a file used exclusively by the operating system.

FILE_ATTRIBUTE_TEMPORARY

A file being used for temporary storage.

The following lists and describes the file attributes that you cannot modify in Windows.

FILE_ATTRIBUTE_COMPRESSED

A compressed file or directory. For files, this means all the data in the file is compressed; for directories, this means that all newly created files and subdirectories are compressed by default.

FILE_ATTRIBUTE_DIRECTORY

A directory.

More information:

chfile Command—Modify Windows File Settings (see page 180)

Windows Account Flags

Flags can be assigned to a user's account to specify particular attributes of that account by using the chusr, editusr, and newusr commands. You can apply more than one flag to each account.

Note: CA Access Control does not require you to enter the complete name of the flag. You can use the shortcuts provided in the table.

Following are the account flags available in Windows.

Shortcut	Flag	Description
blank	UF_PASSWRD_NOTREQD	Indicates that no password is required for the user's account.
cant_change	UF_PASSWORD_CANT_CHANGE	Indicates that the user cannot change the password for the account.
disable	UF_ACCOUNTDISABLE	Indicates the user's account is disabled.
dont_expire	UF_DONT_EXPIRE_PASSWORD	Indicates that the password for this account never expires.
homedir	UF_HOMEDIR_REQUIRED	Indicates the home directory is required. This value is ignored in Windows.
interdomain	UF_INTERDOMAIN_TRUST_ACCOUNT	Indicates a permit to trust account.
lockout	UF_LOCKOUT	Indicates that the user's account is currently locked out; to unlock a locked account, remove this flag

Shortcut	Flag	Description
normal	UF_NORMAL_ACCOUNT	Indicates a default account type that represents a normal user.
notreq	UF_PASSWRD_NOTREQD	Indicates that no password is required for the user's account.
protect	UF_PASSWORD_CANT_CHANGE	Indicates that the user cannot change the password for the account.
script	UF_SCRIPT	Indicates that the login script, which executes disk mapping, is activated when the user starts an application. This flag must be set for LAN Manager 2.0 or Windows.
server	UF_SERVER_TRUST_ACCOUNT	Indicates an account for a Windows NT Backup Domain Controller in this domain.
temp	UF_TEMP_DUPLICATE_ACCOUNT	Indicates a user with an account in another domain; provides access to the domain for this account, but not a trust account.
trust	UF_INTERDOMAIN_TRUST_ACCOUNT	Indicates a permit to trust account.
workstation	UF_WORKSTATION_TRUST_ACCOUNT	Indicates an account for a workstation or server that is a member of this domain.

More information:

chusr Command—Modify Windows Users (see page 186)

Windows Permissions

In the SHARE resource type, you can give access permissions to accessors.

Following are the access permissions available in Windows:

ACCESS_ALL

Permission to read, write, create, execute, and delete resources and to modify their attributes and permissions.

ACCESS_ATTRIB

Permission to modify the resource's attributes.

ACCESS_CREATE

Permission to create a resource, including writing data to it as it's being created.

ACCESS_DELETE

Permission to delete the resource.

ACCESS_EXEC

Permission to execute the resource.

ACCESS_NONE

No access.

ACCESS_PERM

Permission to modify the permissions assigned to a user or an application for a resource.

ACCESS_READ

Permission to read data from a resource and, by default, to execute in the resource.

ACCESS_WRITE

Permission to write data to the resource.

More information:

SHARE Class (see page 463)

Windows Privileges

Windows privileges can be assigned to individual user accounts and groups. Administrators can assign privileges to a user with the chusr or editusr command, or to a group with the chgrp or editgrp command. Users who are added to a group automatically gain all the privileges assigned to the group.

You can use the name of the privilege, or user right, exactly as it appears in the list, or you can add Se to the beginning and Privilege to the end of the name (except for BatchLogon, InteractiveLogon, NetworkLogon, and ServiceLogon, to which you add Right instead of Privilege).

Following are the privileges available in Windows.

Privilege	Default Assignment	Description
AssignPrimaryToken	None	Allows a user to modify the security access token of a process.

Privilege	Default Assignment	Description
Audit	None	Generates security audits.
Backup	Administrators Backup Operators	Allows a user to back up files and directories. This privilege replaces all file and directory permissions.
BatchLogon	None	Allows a user to log in as a batch job.
ChangeNotify	Everyone	Usually, rights to files and subdirectories flow downward; that is, users who do not have rights to a specific directory do not also have rights to access the subdirectories below that directory. This privilege allows a user to access subdirectories, even if that user has no rights to the parent directories.
CreatePagefile	None	Allows a user to create a page file. Security is determined by a user's access to the key: \CurrentControlSet\Control\ SessionManagement
CreatePermanent	None	Allows a user to create special permanent objects, such as \\Device
CreateToken	None	Creates a token object. Only the Local Security Authority can do this. The Local Security Authority ensures that the user has permission to access the system. It is not possible to audit the use of this right. For C2 certification, we recommend that it not be assigned to any user.
Debug	Administrator	Debugs programs or objects such as threads. You cannot audit this privilege. For C2 certification, we recommend that it not be assigned to any user, including system administrators.
IncreaseBasePriority	Administrators Power Users	Allows a user to increase the execution priority of a process.
IncreaseQuota	None	Allows a user to increase the object quotas.
InteractiveLogon	Most groups	Allows the user to log in interactively.
LoadDriver	Administrators	Allows a user to install and remove device drivers.
LockMemory	None	Allows a user to lock pages in the memory of the computer so the pages cannot be automatically backed up on a backing store like PAGEFILE.SYS.
MachineAccount	None	Allows a user to add a new machine to a domain.
NetworkLogon	Everyone	Allows users to connect to a computer from anywhere in the network. This means users do not have to be at a specific place or terminal to log into their computer.

Privilege	Default Assignment	Description
ProfileSingleProcess	Administrators Power Users	Allows a user to use performance-monitoring tools in order to monitor the performance of a single process.
RemoteShutdownPrivilege	Administrators Power Users	Allows a user to shut down a Windows system remotely.
Restore	Administrators Backup Operators	Allows a user to restore backed-up files and directories. This right replaces all file and directory permissions.
Security	Administrators	Allows a user to specify what types of resource access (such as file access) are to be audited, and to view and clear the security log. **Note:** This privilege does not allow the user to set system auditing policies using the Audit command from the Policy menu in Microsoft's User Manager. Administrators always have the ability to view and clear the security log.
ServiceLogon	None	Enables a process to register with the system as a service.
Shutdown	Administrators Backup Operators Everyone Power Users Users	Allows the user to shut down the system from the system console.
SystemEnvironment	Administrators	Allows a user to modify the system environment variables. This enables the user to set up the system environment at their workstation, and ensure that all other users working on the same workstation use the same setup.
SystemProfile	Administrators	Allows a user to perform profiling (performance sampling) on the system.
SystemTime	Administrators Power Users	Allows a user to set the time for the internal clock of the computer.
TakeOwnership	Administrators	Allows a user to become the owner of files, directories, printers, and other objects on the computer. This right replaces all permissions protecting objects.
Tcb	None	Enables a process to perform as a secure, trusted part of the operating system. Some subsystems are granted this privilege.

More information:

chusr Command—Modify Windows Users (see page 186)
chgrp Command—Modify Windows Groups (see page 181)