CA IT Client Manager

encUtilCmd Command Reference

Release 12.8

CA Technologies Product References

This documentation set references to the following CA products:

- CA Advantage® Data Transport® (CA Data Transport)
- CA Asset Intelligence
- CA Asset Portfolio Management (CA APM)
- CA Common Services™
- CA Desktop Migration Manager (CA DMM)
- CA Embedded Entitlements Manager (CA EEM)
- CA Network and Systems Management (CA NSM)
- CA Patch Manager
- CA Process Automation
- CA Business Intelligence
- CA Service Desk Manager
- CA WorldView™
- CleverPath™ Reporter

Contact CA Technologies

Contact CA Support

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

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

Providing Feedback About Product Documentation

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

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

Contents

Chapter 1: Using encUtilCmd

encUtilCmd is a command line utility for the ENC infrastructure. The utility provides the following functionality:

Auditing

Auditing [command: auditapi]

Allows access to the ENC auditing configuration database.

Authorization

Create Rule-set [command: create]

Create a bare set of authorization rules.

Export [command: export]

Read the current authorization rules from common store and write to a structured text file.

Import [command: import]

Read authorization rules from a structured text file and write to the CMS common store.

Import Database [command: importdb]

Read authorization rules from a structured text file and write to the CMS CCNF database.

Verify [command: verify]

Tests authorization rules based on the contents of an authorization rule file and customer provided input.

Authentication

Certificate Import [command: certimport]

Imports certificates and PKCS#12 shrouded keys into the Microsoft certificate store for use by ENC.

Verification [command: certverify]

Tests and enumerates installed certificates: availability and validity.

Miscellaneous

Client [command: client]

Allows you to either enable or disable the ENC Gateway Client.

Netdiag [command: netdiag]

Provides a set of ENC specific diagnostics.

Server Status [command: server]

Displays the status of an ENC Gateway Server.

Updateconfig [command: updateconfig]

Automatically copies policies from the policy or common store to the ENC components.

auditapi—Check ENC Local Auditing Configuration

Short form 'au'

This command provides access to the ENC auditing configuration database on the local machine. Though it is recommended to use the graphical user interface to configure auditing, this command allows you to control some elements of auditing where the GUI is not accessible.

auditapi usage:

```
encUtilCmd auditapi arguments [options]
```

Options:

-c Categories

Displays all audit categories.

-cd Disable Category

Disables one or more categories and allows regular expression for pattern matching.

-ce Enable Category

Enables one or more categories and allows regular expression for pattern matching.

-da Dump All

Dumps all audit records.

-dc Dump Category

Dumps the audit records in the given category.

-e Enabled

Identifies if the message is enabled.

-f Filter

Applies filter using regular expression (audit records).

-ggs Get Global State

Displays current global auditing state.

-md Disable Message

Disables one or more messages and allows regular expression for pattern matching.

-me Enable Message

Enables one or more messages and allows regular expression for pattern matching.

-mgr Manager

Forces manager access to overwrite.

-sgs Set Global State

Specifies the global auditing state.

-test Test Pass

Displays only the categories or messages that matched during pattern matching without committing changes to the configuration database.

where:

-c Categories [no arguments]

Shows all audit categories. This lists the top-level categories and their associates state.

-cd Disable Category [pattern]

Disables one or more categories. The argument specifies a substring of the category name to be matched. For example '-cd er' matches the error and the server. To match only the error, you can fully qualify the expression as '-cd "^error$"'.

Note: The symbol '^' is a special character for the Windows command processor, and must be enclosed within braces to be passed through to the utility command.

-ce Enable Category [pattern]

Enables one or more categories. The argument specifies a substring as per the Disable Category option.

-da Dump All

Dumps all audit records to the console. This option lists all audit record identities, along with their associated categories and individual record states (enabled or disabled).If you want only a subset of the audit records, combine this option with the -f and -dc options.

-dc Dump Category [category]

Dumps only the audit records in the specified category.

Note: Use this option in combination with the -da switch.

Example:

```
encutilcmd au -da -dc
```

This shows only audit records in the error category.

-e Enabled [message-id]

Identifies if a message is enabled. The argument to this option is the full audit record identity such as "IDS_CLI_NO_CERTS". The utility will then display the effective state of the audit record, its category, and reason for the state of the record.

-f Filter [pattern]

Applies filter using regular expression (audit records). This option can be combined with the -da option to show only records that match the specified regular expression.

-ggs Get Global State

Displays the current global auditing configuration state. The possible states can be All, None, or Category.

Use the -sgs option to set the global state.

-md Disable Message [pattern]

Disables one or more messages and allows regular expression for pattern matching. This option disables an individual record.

-me Enable Message

Enables one or more messages and allows regular expression for pattern matching. This option will enable an individual record, though its effective state may be disabled by its parent category.

Use the -e option to show the effective state for an individual record.

-mgr Manager

Forces manager access to overwrite and allows local auditing configuration to be changed even if the configuration is centrally managed by the DSM security policy.

Note: This option requires administrative privileges.

-sgs Set global state

Specifies the global auditing state.

The states are:

- None - All auditing is disabled.
- All - All auditing is enabled.
- Category - Messages are controlled by category and individual state.

Note: If the state is set to Category, use the -e option to see why a message is enabled or disabled.

-test Test Pass

Displays only the categories or messages that matched during pattern matching without committing changes to the configuration database.

You can use this option to evaluate what would change without actually committing the changes.

certimport—Import Certificates into the Local Machine Certificate Store

Short form 'certi'

The certificate import command allows you to import certificates into the Microsoft local machine certificate store for use by ENC. Root certificates will be placed into the "Trusted Root Certification Authorities" folder, and PKCS#12 certificate key-pairs will be added to the "Personal" folder.

The usage output is shown here:

certimport - Verify that the required certificates are available on the current machine.

certimport usage:

```
encUtilCmd certimport arguments [options]
```

Options:

-a Skip Admin

Skip administrators check.

-i PKCS#12

Specify a PKCS#12 shrouded certificate and key to import. Multiple -i options can be specified but must have matching -p entries.

-p Password

Specify the input password for a PKCS#12 file.

-r Root

Specify a root certificate to import. Multiple –r options can be provided.

-v Verbose

Run command with maximum output.

where:

-a Skip Admin

By default, the command will perform a check that you are an active member of the administrators group before executing the import commands. This allows you to skip this check if you have sufficient privileges to perform the imports without being a member of the administrators group.

-i PKCS#12

Specifies the path to the input PKCS#12 file. You can supply multiple '-i' options, but each one must be balanced with a corresponding '-p' option. PKCS#12 files are imported with their private keys marked as non-exportable.

-p Password

This option specifies a password to be used in decrypting the PKCS#12 shrouded file.

-r Root certificate

This option specifies a root certificate to be imported to the Microsoft system store.

-v Verbose

Specifying the verbose options will print more information about the import operations as they occur.

certverify—Verify that the Required Certificates Are Available

Short form 'certv'

Use the certificate verify command certverify -to verify that the required certificates are available on the current machine.

The usage output is shown here:

certimport usage:

```
encUtilCmd certverify arguments [options]
```

Options:

-a Skip Admin

Skip administrators check.

-l Loopback

Attempt a local loopback connectionand report on identities used.

-t Target

Specify the local FQDN for machines that are not fully configured for use in loopback.

-v Verbose

Run command with maximum output.

where:

-a Skip Admin

By default, the command will perform a check that you are an active member of the administrators group before executing the import commands. This allows you to skip this check if you have sufficient privileges to execute the appropriate certificate loading commands without being a member of the administrators group.

-l Loopback

When specified, the utility will create a client and server security context and perform loopback authentication to ensure that the certificates are valid. The certificate identities used will be displayed. This option will only succeed if the installed certificates are intended for use in client and server authentication.

-t Target

When using the loopback option, if the utility can not automatically determine the fully qualified domain name of the local machine, it may be necessary to specify the DNS name with this option.

-v Verbose

Specifying the verbose options will print more information about the import operations as they occur.

create—Create an Authorization Rule-set

Short form 'cr'

This command will create a bare set of authorization rules including an ENC infrastructure realm, a DSM management realm and, optionally, named DSM member realms. The output of the command can further be used in authorization testing, import to the common store or import to the configuration database.

The syntax is as follows:

```
encUtilCmd create arguments [options]
```

Options:

-ex Export XML

Write an XML version of the rules to the specified file.

-o Output

Write output to specified file. The default is to write to the console.

-r Realm

Specify the name of a realm to create. This option can be specified multiple times.

-tb Test Batch File

Create a named batch file to call and test the script created by -ts.

-ts Test Script

Create a named test script. Requires the –o option to be set.

-v Verbose

Run command with maximum output.

where:

-ex Export XML

This will output the rule-set in XML form. This can be used as direct input to the ccnfregdb command.

-o Output

This specifies the output file to use to store the textual version of the rule-set. The default is to write the rules to the console.

-r Realm

The option allows you to specify a named realm for inclusion in the rule-set. Multiple realms be specified by supplying multiple instances of the '–r' option.

-tb Test Batch File

This option instructs the utility to create a test batch file to call the test script created by the –ts option.

-ts Test Script

This option creates a test script to exercise the newly created rule-set.

-v Verbose

Specifying the verbose options will print more information about the rule-set operations as they occur.

export—Export a Set of Authorization Rules from the Configuration Store

Short form 'e'

The export command extracts a current authorization rule set from the DSM common store and converts it to a flat file structure as documented later.

```
encUtilCmd export arguments [options]
```

Options:

-ab Abbreviate

Specify an export rule set, using abbreviated events.

-al Alternate

Specify an alternate rule set.

-ex Export XML

Write an XML version of the rules to the specified file.

-o Output

Write output to specified file.

The default is to write to the console.

-v Verbose

Run command with maximum output.

where:

-ab Abbreviate

Exports the event names as their abbreviated versions.

Example: ServerRegisterAgent is exported as SRA.

Note: The verify -pe command and option combination will display the full event names and their abbreviated forms in braces.

-al Alternate

By default, the export command reads the authorization rule set from the common store location “itrm/common/enc/authz”. To allow for evaluation of rules on a machine without affecting machine operation, it is possible to specify an alternate location – such as “test1” – where to place the rules.

-ex Export XML

During export of the rules, create and save an XML version suitable for use by the ccnfregdb tool.

-o Output

Specifies an output file for the flat file rules. By default, the utility prints the output to the console.

-v Verbose

Executes the export command in verbose mode. Use of the ‘-v’ option is cumulative – it can be specified on the command line one or more times to increment the level of verbosity.

import—Import a Set of Authorization Rules into the Configuration Store

Short form 'import'

The import command reads its input from a specified flat file; validates the input to an extent and then writes to the specified area of the common store.

```
encUtilCmd import arguments [options]
```

Mandatory Arguments:

-i Input File

Specifies the input file.

Options:

-al Alternate

Specify an alternate rule set.

-ex Export XML

Write an XML version of the rules to the specified file.

-fl Flush store

Flush the current rule set from the common store before writing new set.

-lr List Realms

Print the name of the realms from the input file.

-mgr Manager

Force manager access to overwrite.

Requires administrative privileges.

-notify Notify

Send signal to ENC processes to notify that configuration has changed.

-nw No write

Do not commit the rules; only parse the input file.

-v Verbose

Run command with maximum output.

where:

-i Input File

This mandatory argument specifies the name of the input flat file.

-al Alternate

By default, the import command writes the authorization rule set to the common store location “itrm/common/enc/authz”. To allow for evaluation of rules on a machine without affecting machine operation, it is possible to specify an alternate location – such as “test1” – where to place the rules.

-ex Export XML

During import of the rules, create and save an XML version suitable for use by the ccnfregdb tool.

-fl Flush Store

If an existing rule set already exists in common store, the utility will not overwrite the current set to prevent possible corruption or conflicts. Use the ‘-fl’ flush command to clear the existing rule set before committing the new one.

-lr List Realms

This option lists the realms read from the input file.

-mgr Manager

If the current rule set has been applied as a centrally managed policy, and the policy set prevents the machine from communicating due to incorrect rules definition, it is useful to use the ‘-mgr’ switch to force the rules to be applied – overriding the centrally managed policy option. When the machine next receives a policy update, these rules will then be overwritten by the centrally defined ones.

-notify Notify

Use this option to ensure that the ENC Gateway reads the new configuration data immediately. The utility will signal to all listening processes that the authorization configuration has been updated.

-nw No Write

The ‘no write’ option prevents the utility from committing the new rule set to the common store. This allows the utility to load, parse and evaluate the new rule set without having to commit it to the common store.

Note: This option does not stop the operation of the flush command. This makes it possible to use a combination of ‘–fl’ and ‘–nw’ to flush the existing common store without having to write a new rule set. This can be useful for testing empty configuration sets and clearing old test sets.

-v Verbose

Executes the import command in verbose mode. Use of the ‘-v’ option is cumulative – it can be specified on the command line one or more times to increment the level of verbosity.

importdb—Import a Set of Authorization Rules into the Database

Short form 'importd'

The importdb command reads its input from a specified flat file; validates the input and then utilises the ccnfregdb tool to write the rule set to the CCNF database. As the authorization configuration should only be delivered to ENC Gateway nodes, this command will normally create a new policy for subsequent association with a DSM group of ENC nodes. This behavior can be amended by specifying the default option to write to the existing default computer policy.

```
encUtilCmd importdb arguments [options]
```

Mandatory Arguments:

-i Input File

Specifies the input file.

-n Policy Name

Specify the policy name to import to. This must be unique.

Options:

-def Default

Write to the Default Computer Policy.

-desc Description

Defines the policy description.

-ex Export XML

Write an XML version of the rules to the specified file.

-lr List Realms

Print the name of the realms from the input file.

-m Manager

CCNFREGDB: Specify the target Manager name.

Default is localhost.

-nw No write

Do not commit the rules; only parse the input file.

-o Overwrite

CCNFREGDB. Specify whether any existing configuration database values should be overwritten.

Default is not to overwrite.

-s Skip Auto

CCNFREGDB. Specify whether the Automatic configuration update for affected computers should be skipped.

The default is to force immediate update.

-v Verbose

Run command with maximum output.

where:

-def Default

This option specifies that the authorization data should be imported to the default computer policy rather than a new named policy. This will distribute authorization data to all DSM nodes associated with the current DSM domain.

-desc Description [text-string]

This option applies an operator sepcified descriptive string to the imported policy. This is used to add comments about the purpose of the policy.

-ex Export XML

During export of the rules, create and save an XML version suitable for use by the ccnfregdb tools.

-i Input File

This mandatory argument specifies the name of the input flat file.

-lr List Realms

This option lists the realms read from the input file.

-m Manager

This option is passed to the ccnfregdb tool as the target manager of the registration command. By default, the target is 'localhost' as the command is expected to be executed on the domain manager host.

-n Policy Name [text-string]

This option names the newly imported policy. This name should be short, yet descriptive.

Example: ENC-RuleSet-Main

-nw No Write

The 'no write' option prevents the utility from committing the new rule set to the database. This allows the utility to load, parse and evaluate the new rule set without having to commit it.

-o Overwrite

This option is passed to the ccnfregdb tool. It specifies whether any existing rules in the configuration database should be overwritten. The default option for the ccnfregdb tool is to not overwrite any existing data.

-s Skip Auto

This option is passed to the ccnfregdb tool. It specifies that the automatic configuration update of affected computers should be skipped. The default behaviour is to immediately deliver policy to all affected computers.

-v Verbose

Executes the importdb command in verbose mode. Use of the '-v' option is cumulative – it can be specified on the command line one or more times to increment the level of verbosity.

client—Set ENC Client Configuration

Short form `cl´

This command provides the ability to enable or disable the ENC Gateway Client and set other configuration items relevant to the client.

Using this command, you can set the target ENC Gateway Server and its configured ports, and/or set proxy configuration information.

In the normal course of events, the ENC client is configured by policy. This may include the username and password required to connect to an internet proxy server. If, by mistake, the wrong password is sent out by policy, the target machines may be cut off from ENC because they cannot authenticate with the proxy anymore.

If the machines are separated from the Domain Manager by a firewall then they cannot receive a policy update to correct the problem. To address this, you would normally need to manually enter the correct credentials into comstore using ccnfcmda but this is insufficient because the password is encrypted. Encutilcmd provides a means of entering encrypted passwords.

The syntax is as follows:

```
Encutilcmd client [options]
```

Mandatory Options:

-proxy_http

Set the credentials for a HTTP proxy -

-proxy_socks

Set the credentials for a SOCKS5 proxy

-state

Set the state of the ENC Client

State options:

-port_http

Specified the target HTTP port for the ENC Gateway server.

-port_tcp

Specifies the target TCP port for the ENC Gateway Server.

-server

Specifies the fully qualifed domain name of the target ENC Gateway server.

Proxy options:

-password [password]

Specifies the password for the proxy. If this argument is omitted or provided as the '*' character, the command will prompt for it, without echo of what you type.

-proxy_clear

Specifies that one or more of the stored proxy configuration entries should be deleted.

-proxy_host

Specifies the host name for the proxy server.

-proxy_port

specifies the port number for the proxy server.

-proxy_view

Use this option to display the current stored proxy configuration entries.

-user [username]

Specifies the username for the proxy.

Miscellaneous options:

-mgr

If the machine configuration is centrally managed, you must use the option to force the configuration to be changed. This option requires administrative privileges. The changes may be lost the next time centralized policy is delivered.

-restart

When the SSA and CAM are configured to use ENC for the first time they are required to restart. Using this option will apply the restart upon completion of the status command.

-start

Use this option to start the ENC client upon completion of the current command. Can be combined with –stop to cycle the ENC client.

-stop

Use this option to stop the ENC client after the completion of the current command. Can be combined with –start, above.

Examples:

```
Encutilcmd client –state show
```

This will display the current configuration state of the ENC client.

```
Encutilcmd client –state enabled –server example.forwardinc.ca -start
```

This will set the client state to enabled, configure the client to connect to the ENC Gatewat server at example.forwardinc.ca, and start the ENC client.

```
Encutilcmd client –proxy_http -proxy_host <proxy server name> -proxy_port <proxy server port> –user <username> –password <password>
```

This command will set the HTTP proxy connection details for the client.

netdiag—Provides a Set of ENC Specific Diagnostics

Short form 'ne'

Use the netdiag command to get a set of ENC specific diagnostics.

netdiag usage:

```
encUtilCmd netdiag arguments [options]
```

Options:

-r **Resolve**

Attempts to resolve the given name to an ENC address.

-Summary **Summary**

Prints a summary output of the configuration and status of the local ENC components.

-track **Tracking**

Adds the specified security sub-strings to the tracking list on the current machine. All authorization requests containing these strings are traced at a high level.

-trclear **Clear Tracking**

Clears the current tracking values.

-trshow **Show Tracking**

Shows the current tracking values.

-v Verbose

Runs the command with maximum output.

where:

-r Resolve [host-name]

Attempt to resolve the given name to an ENC address.

-summary Summary [no arguments]

Prints a summary output of the configuration and status of the local ENC components.

-track Tracking [text-string(s)]

Adds the specified security sub-strings to the tracking list on current machine. All the authorization requests contain these strings will be traced at a high level.

Note: You can specify multiple strings by using # as the separator.

This allows for selective tracing of targeted machines rather than having to increase the general trace level and include all machines.

Example:

```
-track dc=forwardinc#cn=someone#cn=someone-else
```

This increases the tracing level of authorization requests where the authenticated identity of the client contains one of the sub-strings "dc=forwardinc", "cn=someone", or "cn=someone-else".

-trclear Clear Tracking

Clears the current tracking values.

-trshow Show Tracking

Displays the current tracking values.

-v Verbose

Prints more information about the netdiag operations as it occurs.

updateConfig—Update ENC configuration

There are a few ENC components such as the server, socket adaptor and tomcat that are not directly configured from policy or the common store. Instead they have their own private data files. In the normal course of events, policy is automatically copied to these files, however if, for some reason, you need to set parameters locally, this command will perform the copy operation. A typical example, is that a machine has been misconfigured and is now cut off from the Domain Manager by a firewall. Such a machine cannot receive a new policy to correct the problem which must now be fixed manually. The necessary changes can be made to comstore using the ccnfcmda command and transferred using “encutilcmd updateconfig”.

The updateConfig command will normally notify all processes that all configuration data has been updated. In some cases, it may be more desirable to update only a single area.

Options:

-audit Audit

Notify an update change for auditing configuration.

-authz Authorization

Notify an update change for authorization.

-server Server

Notify an update change for the ENC server process.

-ssa SSA

Notify an update change for the SSA.

verify—Load and Verify a Set of Authorization Rules

Short form ‘v’

The verify command allows for the evaluation of one or multiple events against the criteria defined in an authorization rule set. Validation parameters can be passed directly to the command line to evaluate a single event, or supplied as a script file to evaluate one or more events serially.

```
encUtilCmd verify arguments [options]
```

Mandatory Arguments:

-e Event

Specify an event to evaluate (use -pe for full list). Will require event specific options to be set. This option is ignored if -f is set.

-f Test file

Load and process the specified test file.

-i Input File

Specifies the input file. If not specified, the common store is used directly. If specified, -al should also be specified.

Options:

-al Alternate

Specify an alternate rule set.

-date Date

Specify a date for the test to be evaluated against [format YYYYMMDD].

Default – current date.

-error Error

Pause for a key-press if a rule fails.

-fl Flush store

Flush the current rule set from the common store before writing new set.

-nt NodeType

Specify the node type for the test (MRS/SRS/ROUTER).

-p Principal

Specify the security principal for the test (URI).

-pause Pause

Pause for a key-press after each event.

-pe Print Events

Print a list of event types.

-q Quiet

Do not show rule matches - only event completion summary.

-s SourceIP

Specify a source IP address for the test.

-t Target

Specify the target for the test.

-time Time

Specify a time for the test to evaluated against [format HHMMSS].

Default – current time.

-tz Time Zone

Specify the time zone of the target [format UTC+HHMM].

-v Verbose

Run command with maximum output.

Note: The 'e', 'f', and 'i' options are all listed as mandatory arguments but only '-e' or '-f' need to be specified.

-al Alternate

Specifies an alternate rule set to use for the evaluation. This allows you to write a new rule set into an alternate location for testing (if using the '-i' switch) or use an existing alternate set if already previously imported.

-date Date

Specifies a simulated date that should be used during evaluation. By default the current date is used. This option has a format of YYYYMMDD where YYYY is the 4 digit year, MM is the 2 digit month (01-12) and DD is the 2 digit day (01-31). The date can be partially specified as YYYY or YYYYMM, where the month and days will be assumed to "01" respectively.

-e Event

Specifies the event type to evaluate. Refer to section "Event Types" for a full list of events and required parameters. Executing the verify command with the '-e' option without the required set of parameters will record an error.

-error Error

When evaluating a set of rules, the verification utility will not stop if a rule fails, but will continue to evaluate the remaining rules. This option can be specified to pause the rule evaluation when an error occurs in a rule entry.

-f Test File

Specifies the test script file containing the events to evaluate. Please refer to section "Script Commands" for the available commands for the script.

-fl Flush Store

When the '-i' option is specified, it may be necessary to flush existing rules from the common store before writing a new set. Use this option to remove any existing rules.

-i Input File

Specifies an input rules file set. If not specified, then the current rules in the common store are used explicitly.

-nt Node Type

Specifies the node type to be used during event evaluation. The default is determined from configuration but this option allows you to emulate behaviour as a different node type. The available values are a combination of MRS, SRS and Router. If multiple options are supplied, they should be provided within quotes like so: -nt “SRS Router”.

The “Agent” node type is always implicitly assumed.

-p Principal

Specifies the security principal to be used during event evaluation.

-pe Print Events

This option specifies that the utility should print a list of the recognized event types to the console.

-q Quiet

Do not show the names of the matching rules during event evaluation; only show the summary status.

-s SourceIP

Specifies the source IP address for the ‘ipconnection’ event.

-t Target

This option specifies the target to be used during event evaluation. This is event context specific. Please refer to section “Event Types” for the relevant targets.

-time Time

This option specifies a simulated time to be used during event evaluation. The default behaviour is to use the current time. The format of this option is HHMMSS, where HH is the hour (0-23), MM is the minute (00-59) and SS is seconds (0-59). The time can be partially specified using HH or HHMM where the minutes and seconds will be assumed to be “00” respectively.

-tz Time Zone [time-string]

Specifies the time zone of the target of an operation during the rules verification. The format of this option is UTC+HHMM.

Example: -tz UTC+02:00

-tz UTC-05:00

Each ENC object transfers its configured time zone when registering with the ENC infrastucture. All access control entries against a given target will be evaluated with that time zone. The verification command can simulate the time zone with this option, else assume a time zone of UTC+0.

-v Verbose

Executes the verify command in verbose mode. Use of the '-v' option is cumulative – it can be specified on the command line one or more times to increment the level of verbosity.

Script Commands

This section lists all of the commands available in the script processing section of the verify command.

The verify command accepts either a direct command line or a script file. The script file has a very simple format, like so:

```
Keyword Value(s) <EOL>
```

Any lines beginning with a ';' or '#' are deemed to be comment lines and are ignored.

The available keywords and possible values are:

principal [*variable*]

Specifies the security principal URI that should be performed in the authorization check. The value is freeform and everything from the keyword to the end of the line is loaded as the principal name. This keyword is not validated for correctness.

sourceip [*variable*]

Specifies the IP address to be used in an address authorization check. This is also freeform and everything from the keyword to the end of the line is loaded into the address. This keyword is not validated for correctness.

nodetype [*variable*]

Specifies the node type for the authorization checks. Can be one or more of the following:

- MRS – Master Node (Manager)
- SRS – Slave Node (Server)
- Router – Router Node (Router)

The agent type is always implied.

target [*variable*]

Specifies the target of the operation (the secured object).

date [*variable*]

Specifies the simulated date for the test. The format is the same as the verify command's "-date" option.

time [*variable*]

Specifies the simulated time for the test. The format is the same as the verify command's "-time" option.

clear

Clears all variables or a single specified variable. All keywords marked as [variable] can be cleared.

event [*variable*]

This sets the event type to be tested and also initiates the test evaluation

Example Test Script:

An example test script is shown below:

```
1       #
2       # emulate an ip connection to all purpose ENC node
3       # on 25th December 2008 at 12:34:04
4       #
5       nodetype mrs srs router
6       sourceip 192.168.0.1
7       date 20081225
8       time 123404
9       event ipconnection
```

This script sets the "nodetype", "sourceip", "date" and "time" variables and then performs an "ipconnection" access check.

Example Test Output:

The example test output is shown below:

```
Handling event <ipconnection> from line <9>
Date <20081225> Time <123404> Node-type <mrs srs router>.
  Operation was allowed for reason
        'Item found in allowed list by pattern match'
  Pattern:
        '.+'
Event <ipconnection> was <allowed>.
```

Event Types

This section lists the event keywords applicable to the verify command.

In all cases, the relevant parameters are passed to the authorization APIs which will subsequently determine if access is allowed or denied. Why an event was allowed or denied will be displayed, referring to the reason and related access control entry.

Event: **NetworkConnection**

Parameters: **sourceip, nodetype**

Relevant Context: **all nodes**

This event is called during the initial TCP/IP connection to an ENC node. Access to the target node is determined by the authorization white list of addresses.

Event: **AuthenticatedConnection**

Parameters: **principal, nodetype, target**

Relevant Context: **all nodes**

This event is called after a client connection has completed its authentication sequence. The authenticated identity (principal) and the target (realm) are passed to the authorization API.

Event: **ManagerRegisterServer**

Parameters: **principal, nodetype, target**

Relevant Context: **Gateway Manager**

This event is called when an SRS node sends a registration command to an MRS.

Event: **ServerRegisterRouter**

Parameters: **principal, nodetype, target**

Relevant Context: **Gateway Server**

This event is called when a Router node sends a registration command to an SRS.

Event: **ManagerRegisterRouter**

Parameters: **principal, nodetype, target**

Relevant Context: **Gateway Manager**

This event is called when the SRS forwards a router registration command to an MRS.

Event: **ServerRegisterAgent**

Parameters: **principal, nodetype, target**

Relevant Context: **Gateway Server, Router**

This event is called when an end client attempts to register to an SRS or Router node.

Event: **ManagerRegisterAgent**

Parameters: **principal, nodetype, target**

Relevant Context: **Gateway Manager**

This event is called when an SRS forwards a client registration event to the MRS.

Event: **ManagerNameLookup**

Parameters: **principal, nodetype, target**

Relevant Context: **all nodes**

This event is called when an end node performs a name lookup request.

Event: **AgentConnect**

Parameters: **principal, nodetype, target**

Relevant Context: **Gateway Manager, Server**

This event is called when an end node attempts an active outgoing connection to a target node.

Event: **RouterAgentConnect**

Parameters: **principal, nodetype, target**

Relevant Context: **Gateway Router**

This event is called when an end node attempts to connect to a router to set up a virtual connection to a peer node.

Event: **ManagementAccess**

Parameters: **principal, nodetype, target**

Relevant Context: **Gateway Manager**

This event is called when an end node requests management information from the ENC Gateway Manager.

Rules File Format

This section defines the format of the authorization rules file. The following extract shows the basic layout of the file.

The [authz] section title and version keys must be the first entries in the file.

The individual sections (timeRange, timeacl, URIMapping, dnsmapping,and IPAddWhiteList) do not all need to be present or in any predetermined order though they must all be terminated with an “end” keyword if they are present.

The ordering of records within each section is important and will be honoured.

The symbol +* at the end of some lines indicates that there may be multiple entries within a section and should not be entered in a real file.

```
[authz]
RulesVersion=n

realm
{ name xxx notes yyy}+*
end

timeRange
{name xxx enabled x hours hhh type ttt weekdays www year yyyy month mmm day ddd}+*
end

timeacl
{name xxx enabled x SecPrincType xxx events xxx RuleType xxx TimeRange xxx SecPrinc
xxx SecObj xxx SecObjType xxx }+*
end

URIMapping
{URI xxx enabled x type xxx realm xxx}+*
end

IPAddWhiteList
{IPAddress enabled x xxx type xxx}+*
end
```

A detailed breakdown of each section and entry follows.

For all record types - for any value that contains spaces, the value should be enclosed with double quotes to delineate the start and end of the value; for example:

```
name “this is the record name”
```

Additionally, each record must begin and end with an opening and closing brace ‘{‘ and ‘}’.

Common attributes:

There are some attributes that are used across two or more sections. These are detailed here.

Attribute: ‘enabled’

Set to 1 to enable the individual record or to 0 to disable it.

Disabled records will still be processed, validated and stored but will not be acted upon by the authorization component.

Attribute: ‘name’

A readable name to describe the timeRange entry or timeacl entry. This name is not used for direct authorization purposes but is used for diagnostic purposes and for cross-referencing an access control entry with a time-range. For example, a timeRange entry for the 4th July may be called “Independence Day”.

Section: Realm

This section defines realm names for subsequent sections.

Attribute: ‘name’

This defines the name of a realm for subsequent use. Usually, the realm name should be something meaningful to the membership list.

Attribute: ‘notes’

This attribute provides the ability to add notes for this realm entry. The notes are not interpreted in any way, but will be displayed in the authorization configuration graphical user interface.

Section: timeRange

This section defines time ranges for subsequent use by the timeacl section.

Time ranges can either describe generic weekday ranges or a significant date, such as holidays, scheduled maintenance etc.

The first attribute-value pair in every timeRange entry should be the name attribute.

Attribute: ‘name’

As described in common attributes.

Attribute: ‘enabled’

As described in common attributes.

Attribute: 'hours'

This attribute describes the hours that the time-range is in effect. The granularity of the 'hours' entry is in 30-minute periods. The 'hours' value can be specified as a single entry, a range with start and end, or multiple ranges.

- An 'hours' value of "7:00" would mean from 7 a.m. through to 7:29:59 a.m. local time.
- An 'hours' value of "9:00 – 17:00" would mean from 9 a.m. through to 17:00:00 local time.
- An 'hours' value of "9:00 – 11:30 13:00 – 17:00" would mean from 9 a.m. through to 11:30:00 a.m. and 1 p.m. through to 5:00.00 p.m. local time.

Attribute: 'type'

The 'type' attribute determines the kind of time-range record this is. It has possible values of 'normal' or 'special'.

- The 'normal' value defines a normal record that uses the 'weekdays' values to determine when it is active. If the type is 'normal' then the 'weekdays' attribute-value pair must be present.
- The 'special' value defines a record that is only applicable to a specific date. If the type is 'special', then the year, month and day attribute-value pairs must be present.

Attribute: 'weekdays'

The 'weekdays' attribute defines the periods when a normal time-range is active. The value can be a single day, multiple individual days, a day-range or multiple day-ranges.

- A 'weekdays' entry of "Sunday" would define a time-range that is only enabled on Sundays.
- A 'weekdays' entry of "Monday Friday" would define a time-range that is only enabled on Monday and Friday.
- A 'weekdays' entry of "Sunday-Tuesday" would define a time-range that is enabled from Sunday through Tuesday.
- A 'weekdays' entry of "Sunday-Monday Wednesday-Thursday Saturday" would define a time range that is enabled from Sunday through Monday, Wednesday through Thursday, and Saturday (not operative on Tuesdays and Fridays).

Attribute: ‘year’, ‘month’, ‘day’

These attributes are gathered together as they must all be present for a ‘specialdaytype'.

- The ‘year’ value is a four-digit numeric representation of the desired year.
- The ‘month’ value is a numeric value representing the month, with valid values being from “1” to “12”, referring to January through December respectively.
- The ‘day’ value is a numeric value representing the day from “1” thru “31” inclusive.

If any of the special day values are zero (0), then this is used as a wild card match where 0 matches any year, month or day respectively.

Section: timeacl

This section defines individual Time Access Control Entries (TACE’s) to form a Time ACL (Time Access Control List).

The first attribute-value pair in every timeacl entry should be the name attribute.

Attribute: ‘name’

As described in common attributes.

Attribute: ‘enabled’

As described in common attributes.

Attribute: ‘SecPrincType’

This attribute value defines the type of matching to be performed when using this TACE. It can be a value of either ‘normal’ , ‘pattern’ or ‘realm’.

- When set to ‘normal’, the TACE will only be active for the identity that exactly matches the ‘SecPrinc’ attribute value.
- When set to ‘pattern’, the TACE will be active for all identities that match the regular expression in the ‘SecPrinc’ attribute value.
- When set to ‘realm’, the TACE will be active for all identities that match the regular expression in the ‘SecPrinc’ attribute value.

Attribute: ‘Events’

This attribute defines the event or events that this TACE is applicable to. The event names are those as defined in the section “Event Types”.

Attribute: 'RuleType'

This attribute defines the type of TACE this is. The value is either 'allow' or 'deny'.

Attribute: 'TimeRange'

This is a cross reference to a time-range entry. This controls when the TACE is active. The attribute value should match the time-range name exactly. The import utility will validate that each TACE references a valid time-range (though it does not enforce the 'enabled' flag for a time-range).

Attribute: 'SecPrinc'

This is either the explicit name of a security principal, a regular expression used to match multiple security principals, or a realm name.

Attribute: 'SecObj'

This is either the explicit name of a secured object (the target of an operation) or a regular expression used to match multiple secured objects.

Attribute: 'SecObjType'

This defines the type of string in 'SecObj' attribute.

The current defined types are

- 'normal'
- 'pattern'
- 'realm'

Section: URIMapping

This section defines how security principal names are mapped into realms. The first attribute-value pair in each entry must be the 'URI' keyword.

Attribute: 'URI'

This string is used for comparison with the security principal identity to decide realm membership.

Attribute: 'enabled'

As described in common attributes.

Attribute: ‘type’

This attribute defines the type of the entry. It can have a value of ‘normal’ or ‘pattern’.

- When set to ‘normal’ the URI value is used as an exact match for realm membership.
- When set to ‘pattern’, the URI value is used as a regular expression for pattern matching.

Attribute: ‘realm’

This attribute defines the name of the realm or realms that the security principal is mapped to.

Multiple realms can be specified by separating each realm name with a ‘#’ symbol.

There is a special realm name of “*” which matches all realms. Any principal that has a ‘*’ realm association will be able to access all realms. The ‘*’ realm membership should only be used in rare circumstance and only when you fully understand the security implications.

Section: IPAddWhitelist

The section defines the list of IP addresses that are allowed to connect to the target node.

The first attribute-value pair in each entry must be the ‘ipaddress’ keyword.

Attribute: ‘ipaddress’

This string defines an explicit IP address or a regular expression to pattern match against the incoming IP address.

Attribute: ‘enabled’

As described in common attributes.

Attribute: ‘type’

Defines the type of this white-list entry; its value can be either ‘normal’ or ‘pattern’.

- When set as ‘normal’, the IP address must be an exact match to be allowed to continue.
- When set to ‘pattern’, the IP address must pattern match to be allowed to continue.

server—Display ENC Gateway Server Status

This command displays the status of an ENC Gateway Server. It is also used to show:

- The list of clients and servers registered with the Gateway Manager.
- The list of connections currently being handled by a Gateway Router.
- The ENC topology and other options.

The command also provides options to filter or send the output to a file.

The syntax is as follows:

```
Encutilcmd server [query option] [arguments] [-host <host name>] [-port <n>] [-v]
[-dump <file>]
```

Query Options

-client [[-fqdn <host name>] [-pattern]]

Queries the server for a list of registered clients.

-connection

Queries a router for the list of active ENC connections.

-general

Queries a server for its status.

-router

Queries a server for the list of registered Gateway Routers.

-server

Queries an ENC Gateway Manager for the list of registered servers.

-topology

Queries an ENC Gateway Manager for the ENC network topology.

Arguments

-dump <file name>

Writes the command output to an XML file. This can be displayed in a web browser, text editor, or processed by other programs.

-fqdn <host name>

Specifies a host name or pattern to filter the output from -client.

Note: If -pattern is specified, then the value of -fqdn is a pattern otherwise it is a host name.

-host <host name>

Specifies the fqdn of the Gateway Server to be queried.

Default: Local host

-pattern

Specifies the value following -fqdn as a regular expression to be matched with the list of clients.

Default: fqdn is a match to a single computer.

Note: The regular expression syntax follows the Perl standard.

-port <n>

Specifies the port number of the server on the target machine.

Default: 443

-v

Specifies the verbose output if possible.

where:

-client

This option retrieves the list of ENC clients (encClient.exe) registered with the specified server.

If the target server is an ENC Manager, all the clients are returned, and if the target server is an ENC Server, then just the clients registered with that server are returned.

Note: Routers do not return anything.

For a large number of clients, you can send the output to a file using the -dump argument or filter the output using a pattern match.

For each client, the output lists the following information:

- Fully qualified domain name
- Host UUID
- ENC virtual address
- Time zone

Examples: List All Clients

- This example lists all clients on the ENC Server running on the local host.

```
encutilcmd server -client
```

- This example lists all clients on the ENC Server running on the mymgr machine.

```
encutilcmd server -client -host mymgr
```

- This example lists all clients on the ENC Server running on the mymgr machine which is residing in the domain fred.com.

```
encutilcmd server -client -host mymgr -fqdn .*\.fred\.com -pattern
```

- This example shows information about the ENC client running on host1.fred.com.

```
encutilcmd server -client -host mymgr -fqdn host1.fred.com
```

-connection

This option queries a Gateway Router for the list of active connections.

For each connection, it shows:

- The host that made the connection.
- The host that accepted the connection.

As the router is transmitting data between these hosts, for each host it displays the following:

- The real IP address
- The certificate used by that host to authenticate. The certificate description shows the fqdn of the host.

-general

This option displays the general information of the server. This option is also the default query if no query is specified.

The general information includes:

- Server type (manager, server, router, or all)
- Version ID
- The time period that the server has been running for.
- The port number open for ENC connections.
- The fqdn of the ENC server it is registered with. (ENC Server or Manager)
- The number of servers registered with it.
- The address range it uses to allocate virtual ENC addresses to clients in case the specified host is an ENC Gateway. Manager.

Examples: Show General Information of the Server

- This example shows the general information of the server on the local host.

```
encutilcmd server
```

- This example shows the general information of the server on the host myhost.xyz.com.

```
encutilcmd server -general -host myhost.xyz.com
```

-router

Displays the list of Gateway Routers that are registered with the target server.

For each router, it shows:

- The certificate url used by the router to authenticate itself with the manager.
- Server type (Server, router, or both)
- The port number open for client connections
- Version ID

The following example shows the routers registered with mymgr.xyz.com.

```
Encutilcmd server -router -host mymgr.xyz.com
```

-server

This option displays the list of Gateway Servers that are registered with the target server. If this option is used with a Gateway Manager, all the servers are shown. And when it is used with a Gateway Server, all routers registered with that server are shown. The information displayed is the same as the one for the -router option.

The following example shows the servers registered with mymgr.xyz.com.

```
Encutilcmd server -server -host mymgr.xyz.com
```

-topology

This option displays the topology of the ENC network. This shows the Gateway Manager first followed by an indented list showing servers and routers.

For each server, it shows the servers that register with as follows:

Manager

Server1 (This server registers with the Manager)

Server2 (This server also registers with the Manager)

Router1 (This router registers with Server2)

Router2

Server3

Router3

Router4 (This router registers with the Manager)

The following example shows the topology for ENC Gateway Manager mymgr.xyz.com

```
Encutilcmd server -topology -host mymgr.xyz.com
```

Index