# Table of Contents

Getting Started.................................................................................................................. 5  
Intended Audience............................................................................................................. 5  
Overview ............................................................................................................................. 5  
Document Conventions....................................................................................................... 5  
Installing the Enhanced PERL API.................................................................................... 6  
Installation Requirements................................................................................................. 6  
Installation Steps................................................................................................................. 6  
PERL Documentation........................................................................................................... 8  
Examples ............................................................................................................................... 8  
Installing the Legacy PERL API......................................................................................... 9  
Operating Systems.............................................................................................................. 9  
Windows Installation ......................................................................................................... 9  
  Step One: Installing NCM.................................................................................................. 9  
  Step Two: ActivePerl for Windows......................................................................................... 10  
  Step Three: Testing the Installation .................................................................................... 10  
Solaris and Linux Installation ............................................................................................ 10  
  Step One: Installing NCM .................................................................................................. 10  
  Step Two: Installing the PERL Inline-0.33 Module ...................................................... 11  
  Step Three: Testing the Installation .................................................................................... 11  
Relationship between the API and the CLI or Telnet/SSH Proxy ..................................... 11  
Legacy PERL API Functions............................................................................................... 12  
  true_create ()................................................................................................................ 12  
  true_open ....................................................................................................................... 12  
  true_close....................................................................................................................... 13  
  true_exec ....................................................................................................................... 13  
  true_getValue ................................................................................................................ 13  
  true_getText .................................................................................................................. 13  
  true_getColumnCount .................................................................................................... 13  
  true_getColumnName ........................................................................................................ 14  
  printAllNamesAndValuesInResultSet .......................................................................... 14  
Programming Example .................................................................................................... 14  
Commands ......................................................................................................................... 17
Getting Started

Intended Audience

This document is intended for network engineering professionals who:

- Write scripts to automate device configuration.
- Are comfortable with basic Practical Extraction and Reporting Language (PERL) programming, and have an understanding of database schema and access methods.
- Have knowledge of the CiscoWorks Network Compliance Manager (NCM) CLI. NCM CLI documentation is available in Appendix C or can be accessed within the CLI using the `help` command. Most information that is available from the NCM Web interface and the NCM CLI is also available through the PERL API.
- Integrate various third party systems with NCM 1.3, such as network management, workflow, and trouble ticketing solutions.

Overview

This guide includes information on the PERL Application Programming Interface (API). The PERL API enables NCM to communicate with external systems and vice-versa. The PERL API can be used to add and retrieve data to and from NCM.

Common tasks, such as adding devices into NCM and alerting third party systems when a device configuration changes, can be programmatically accessed using the PERL API. Users who want to use other languages can automate their common functions using CLI or Telnet protocols.

NOTE: To install the enhanced PERL API, refer to “Installing the Enhanced PERL API” on page 6.

Document Conventions

This document uses the following conventions:

- File names, directory names, and answers/arguments supplied by the user are represented in Courier font, for example: `ONAAPI.zip`
- Display of on-screen activity is represented in Courier font, for example: `Volume Serial Number`
Installing the Enhanced PERL API

The following modules are provided on the Distribution CD:

- Opsware::NAS::Util
- Opsware::NAS::Client
- Opsware::NAS::Connect

Installation Requirements

PERL version 5.8 or later is required.

If you are using the Auto Installer, skip to the “Auto Installer Method” section in the Installation Steps section below. (NOTE: NCM 1.1 and above does not currently support the system call to run the install.pl script. At this time, you will have to manually install the enhanced PERL API.)

If you are manually installing the PERL API, confirm that certain versions of PERL and/or PERL modules (that are not part of some core PERL distributions) are installed before you begin. Refer to the META.yml file within each package/tarball for its requirements.

If your PERL distribution does not contain all of the required PERL modules, they are available at http://www.cpan.org and/or via PPM. (If you are using ActivePerl, try PPM first.)

To install any of the required modules, use one of the following commands:

- ppm install SOAP-Lite
- cpan install SOAP::Lite

NOTE: that PPM (ppm.exe) is part of the ActivePerl distribution. If you are using ActivePerl, it is recommended that you use the PPM method. You can also run PPM without arguments and then issue the install command. You may need to do this for some PERL modules that have multiple versions to choose from, followed by install # (where # is the item in the list returned by the install command). Keep in mind that PPM prefers to use the '-' as a namespace separator in place of the PERL '::' separator.

NOTE: NMAKE.EXE is installed when installing NCM on a Windows platform. It is located the /client directory. CPAN is simply a wrapper for the perl -MCPAN -e shell command. The CPAN command (or cpan.exe) is part of the core PERL install on all PERL versions since 5.8.0 (including ActivePerl).

Installation Steps

There are two methods for installing the PERL API modules. The first and by far the easiest method is to use the Auto Installer. You can only use the Auto Installer, however, if you have installed the PERL API distribution via the Cisco NCM installer. Otherwise, you must use the manual installation method.
Auto Installer Method:
The Auto Installer installs all of the Opsware::NAS modules as well as their dependencies.

1. Open a shell. If you are on a Windows platform, open a command shell. If you are on a Linux or Solaris platform, you can either open a command shell or SSH into the NCM server. (Note: You will need to have privileges to both create and modify files for NCM as well as PERL. As a result, you might need Administrator privileges on a Windows Platform and root privileges on Linux or Solaris platforms.)

2. Change to the directory where NCM is installed. This directory will have been set when you installed NCM.

3. To run the install script, enter: perl client/perl_api/har/install.pl

NOTE: If PERL is not in your path and/or you have multiple PERL versions installed, use the full path to the PERL executable that you will be using. This should also match the value for the PERL interpreter set in the NCM server configuration.

As noted, the above procedure installs all of the Opsware::NAS modules, as well as their dependencies. However, only "pure perl" dependencies are provided. For example, SOAP::Lite is provided, which includes a minimalist lightweight XML parser. For the best performance, it is recommended that you have the XML::Parser module installed.

If you are using ActivePerl (with a PERL version of 5.8 or better), the XML::Parser module is included with the distribution. Otherwise, you will need to use PPM, CPAN, or manually download and install the module.

Manual Install Method:
Keep in mind that the installation could fail if your PERL installation does not meet certain requirements. Refer to the “System Requirements” section on page 9. In addition, the Opsware::NAS PERL modules are distributed as compressed tarballs, similar modules on CPAN. They are located in the following directory:
<NCM_ROOT>/client/perl_api/Opsware/.

To untar and uncompress all of the modules at one time, you can use the ptar command. ptar is distributed as part of the popular PERL module Archive::Tar, which is included in the standard ActivePerl distributions. To view the contents of the directory and to extract the contents into your current directory, enter: ptar -xzvf PATH/TO/whatever.tar.gz.

For each of the following modules, uncompress and untar the module(s) and change to the directory that was created:

- Opsware::NAS::Util
- Opsware::NAS::Client
- Opsware::NAS::Connect

PERL API Reference Guide
To install the PERL API on a Windows platform with ActivePerl (or any platform running a version of PERL that has the Module::Build module installed):

- perl Build.PL
- perl Build build
- perl Build test
- perl Build install

You may also use the traditional CPAN method. Enter:

- perl Makefile.PL
- make
- make test
- make install

NOTE: If you are using the CPAN method on a Windows platform, you will need to enter nmake rather than make.

**PERL Documentation**

After installing the PERL API, you can view the following PERL POD pages:

- perldoc Opsware::NAS::Client
- perldoc Opsware::NAS::Connect
- perldoc Opsware::NAS::Client::4_5_x
- perldoc Opsware::NAS::Client::6_0_x

Your PERL distribution can also build HTML files for the documentation.

**Examples**

There are PERL API examples in the demo directory. These examples illustrate how to use the PERL API. Keep in mind that it is possible to run the examples without installing the PERL modules by remaining in the demo directory and supplying the relative (or full) path to each example, as in:

- unix_box$ perl demo/list_users.pl
- C:\Windows\Box> perl demo\list_users.pl
Installing the Legacy PERL API

To use the NCM PERL API, you should have at least 50 MB of free disk space (the minimum required disk space 44 MB), and the following installed on your system:

- ActivePerl, Version 5.6 or higher. ActivePerl can be downloaded from http://www.activestate.com/Products/Language_Distributions/
- NCM Client utilities installed and configured
- WinZIP or an equivalent archiving utility

Operating Systems

The NCM PERL API has been tested with the following operating systems:

- Windows 2000 Professional with Service Pack 2
- Windows 2000 Server with Service Pack 2
- Windows 2003 Server
- Windows XP Professional
- Red Hat Linux Enterprise AS (update 2 and 3)
- Solaris 9.x

Note: If you plan to use the PERL API to write scripts to run as Advanced Scripts in NCM, make sure NCM is configured with the correct path to PERL.

Windows Installation

Step One: Installing NCM

You must install the NCM Client on the host on which you are using the PERL API. (Note: You do not have to do Steps 1 and 2 if you are installing the PERL API on the same server on which you installed NCM.)

1. Insert the NCM installation CD into your CD drive. The InstallAnywhere Self Extractor opens. Follow the Install Wizard instructions. (Note: A license key is not required for installing the NCM client.)

2. On the “Choose Install Set” page, select the Client Only option. When prompted, ensure that the hostname of the NCM server is entered correctly. If ActivePerl is not installed, refer to Step Two below. (Note: To verify that ActivePerl is installed, at the command line enter: perl -v and see what prints out.)

3. To verify that the NCM client has been installed, run the <NCM_ROOT>\client\runclient.bat command. You might be prompted for your username and password. Enter the same credentials that you would use to login to the NCM server. Try some commands, such as list device and list user. (Note: <NCM_ROOT> is the drive and directory on which you installed the NCM client.)
**Step Two: ActivePerl for Windows**

ActivePerl for Window should already be installed on your Windows system. *(Note: ActivePerl is not provided on the NCM install CD or with the NCM product.)* Keep in mind that the NCM PERL API must be copied to the appropriate location.

1. Run the `setup_perl.bat` file. The `setup_perl.bat` file is located in `<NCM_ROOT>/client/sdk`.
2. Ensure that the PATH is set correctly to include the directory containing perl.exe (for example, `C:\Perl\bin`). Recent versions of the distribution set the PATH environment variable correctly. The directory where PERL is installed is referred to as `PERL_HOME_DIR` (e.g., `C:\Perl\bin` in the example).
3. Enter `perl -V` at the command prompt to see what your Perl `@INC` variable is set to. *(Note: <NCM_ROOT> is the drive and directory on which you installed the NCM server.)*
4. Copy the `TrueControlAPI.pm` file from `<NCM_ROOT>/client/sdk` to somewhere in the Perl `@INC` path, for example: `C:\Perl\lib`.

**Step Three: Testing the Installation**

To test the PERL API installation

1. Edit the following variables in the `getUserInfo.pl` file (<NCM_ROOT>/client/sdk/examples/perl/getUserInfo.pl). Be sure to remove the leading and trailing `@` on the variable definition.
   - Host
   - User
   - Password
2. Run: `perl getUserInfo.pl`

**Solaris and Linux Installation**

**Step One: Installing NCM**

You must install the NCM Client on the host on which you are using the PERL API. *(Note: You do not have to do Steps 1 and 2 if you are installing the PERL API on the same server on which you installed NCM.)*

1. Insert the NCM installation CD into your CD drive. The InstallAnywhere Self Extractor opens. Follow the Install Wizard instructions. *(Note: A license key is not required for installing the NCM client.)*
2. On the “Choose Install Set” page, select the Client Only option. When prompted, ensure that the hostname of the NCM server is entered correctly.
3. Login to NCM using the CLI login (Telnet) and ensure that the NCM client (`runclient.sh` and `truecontrol-client.jar`, located in `/<NCM_ROOT>/client`) is running. Try some commands, such as `list device` and `list user`. You may be prompted for your username and password. Enter the same credentials that you used to login.
**Step Two: Installing the PERL Inline-0.33 Module**

1. Copy the TrueControlAPI.pm file to somewhere in INC path, for example: /usr/lib/perl5/site_perl. TrueControlAPI.pm is located in /<NCM_ROOT>/client/sdk/.

2. Enter `perl -V` to see what your PERL installs INC variable is set to.

3. Install the PERL Inline-0.44 module. Enter:

   ```
   # cd <NCM_ROOT>/client/Inline/Inline-0.44
   # perl Makefile.PL
   # Do you want to install Inline::C?[n]n
   # make
   # make install
   ```

4. Install the PERL Inline-Java-0.33 module. Enter:

   ```
   # cd <NCM_ROOT>/client/Inline/Inline-Java-0.33
   # perl Makefile.PL
   # Do you wish to build the JNI extension? [yn] n
   # make
   # make install
   ```

**Step Three: Testing the Installation**

To test the PERL API installation:

1. Edit the following variables in the `getUserInfo.pl` file: (/<NCM_ROOT>/client/sdk/examples/perl/getUserInfo.pl)

   - Host
   - User
   - Password

2. Run: `#perl getUserInfo.pl`

**Relationship between the API and the CLI or Telnet/SSH Proxy**

`Session.exec` is used to send a request to the API. The commands accepted by `Session.exec` are, with the exceptions noted below, syntactically identical to those accepted by the CLI or the Proxy interface interactive mode. You may find it convenient to test commands intended for your programs using Telnet to your server and entering the commands manually.

All commands accepted by the CLI or Telnet/SSH Proxy are valid for `Session.exec`, except for the `show version`, `import`, and `help` commands. The PERL API does not support these.
Legacy PERL API Functions

An '@' and '$' symbol preceding the function name signifies the type of data returned by the function. Each function can be called directly by its name. The PERL module exports the function name into the calling programs namespace.

true_create()
Arguments: None
Returns: SessionObject
Description: SessionObject includes the following methods:

- String getOption(optionName, defaultValue)
- Void open(userName, password, NCMHostUrl)

Note: NCMHostUrl is a string that has the format <hostname>:<portNumber>

- Result exec(command)
- String getTimestamp(JavaDateObject)
- int getShort(JavaShortValue)

true_open
Arguments: username, password [, NCMHostUrl [, SessionObject]]
Note: Arguments in square brackets are optional.
Returns: Undefined
NCMHostUrl defaults to localhost:1099 (if not specified).
SessionObject defaults to the session opened by true_open() (if not specified).
Description: Authenticates against the NCM server with username and password. If the authentication fails, it prints an error. This is a high-level API call for writing simple scripts. To capture a failed authentication, use the following API calls:

```perl
my $session = true_create();
eval {
    $session->(username, password [, NCMHostUrl]);
}
if ($@) {
    # handle authentication failure here
    if (caught("java.lang.Exception") {  
        # grab $@->getMessage() or $@->toString()
        # to get the Java error, which may not always
        # be bad password. Could be host not reachable
        # or something else.
    } else {
        # something happened in Perl, not in Java
        # that caused the failure. Print $@
    }
}
```
true_close
Arguments: [sessionObject]
Returns: Unspecified
Description: SessionObject defaults to the session opened by true_open() (if not specified). Disconnect from the NCM Server.

true_exec
Arguments: command [, sessionObject]
Returns: ResultObject
Description: SessionObject defaults to the session opened by true_open() (if not specified). Executes the command in the session. true_exec() returns a ResultObject. ResultObject has these methods:

  String getReturnStatus()
  Boolean getSucceeded()
  String getText()
  ResultSet getResultSet()
  String getStackTrace()
  ResultSet is a java.sql.ResultSet

true_getValue
Arguments: ResultSet, ColumnName
Returns: Depends on ColumnName
Description: Grabs the value in a named column from the current row in the ResultSet.

true_getText
Arguments: ResultObject
Returns: text string
Description: Grabs the text string from the result object.

true_getColumnCount
Arguments: ResultSet
Returns: int
Description: Returns the number of columns in the ResultSet.
true_getColumnName
Arguments: ResultSet, columnNumber
Returns: Depends on columnNumber.
Description: Grabs the column name associated with the column number.

printAllNamesAndValuesInResultSet
Arguments: ResultSet
Returns: unspecified
Description: Prints the column count and all the column names to STDOUT. This is useful for debugging scripts.

Programming Example
This section shows how a simple application can be written using PERL.

Note: A ResultSet refers to a list of all rows in the database that match a SQL Statement. You can read the results of each row using an iterator such as next().

In general, to find the phone number of a person in a database that contains first names, last names, and phone numbers for a large number of people, you would typically connect to a database and issue a SQL Statement. The results (the names and numbers that match the SQL Statement, for example all users with the first name Bob), would be available within a ResultSet. If several entries in the database correspond to individuals with a first name of Bob, you can read each row and check for the last name using the iterator next().

In terms of NCM, the List Device command lists all of the devices currently managed by NCM and displays information for these devices. The following data is a subset of the data stored for each device.

<table>
<thead>
<tr>
<th>Database Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PrimaryIPAddress</td>
<td>Displays the device’s IP address.</td>
</tr>
<tr>
<td>PrimaryFQDN</td>
<td>Displays the fully qualified Domain Name, for example POS6-0.BR1.SEA1.ALTER.NET.</td>
</tr>
<tr>
<td>DeviceName</td>
<td>Displays an internal name for the device, for example L2-Switch-Bld3 Closet.</td>
</tr>
<tr>
<td>Vendor</td>
<td>Displays the device’s manufacturer, for example Nortel Networks.</td>
</tr>
</tbody>
</table>
The following program retrieves the above device data.

```perl
use TrueControlAPI;
use strict;

my $username  = "@FILL_IN_USERNAME@";
my $password  = "@FILL_IN_PASSWORD@";
my $CiscoHost = "@FILL_IN_HOST@";
true_open($username, $password, "$CiscoHost:1099");
my $res = true_exec("list device");
my $resultset = $res->getResultSet();
while($resultset->next() )
{
    print(true_getValue($resultset,"PrimaryIPAddress"), "\n");
    print(true_getValue($resultset,"PrimaryFQDN"), "\n" );
    print(true_getValue($resultset,"DeviceName"), "\n" );
    print(true_getValue($resultset,"Vendor"), "\n")
}
true_close();
```
Copy the above program into a file and save it as ListDevice.pl.

Run the program using the command: `perl ListDevice.pl`. The program prints a list of all devices registered with NCM. For each device, the IP address, fully qualified Domain Name, Device Name, and Vendor are printed, if available.

<table>
<thead>
<tr>
<th>Statements</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use TrueControlAPI</td>
<td>Indicates the module that is to be loaded by PERL.</td>
</tr>
<tr>
<td>true_open ($username, $password, $server:$port)</td>
<td>Enables you to create a session. A session must be created for any command to be run.</td>
</tr>
<tr>
<td>true_exec (&quot;list device&quot;)</td>
<td>Enables you to run the List Device command. Refer to the help command in the proxy for information on commands and their arguments.</td>
</tr>
<tr>
<td>true_close ()</td>
<td>Closes the session.</td>
</tr>
</tbody>
</table>

**Note:** Several examples are provided in the `c:\<NCM_ROOT>\client\sdk\Examples\perl` directory.
Commands

This section provides information for issuing commands and receiving the correct result data types. When invoked via the NCM PERL API, the required user permissions for all commands are the same as for the Telnet/SSH Proxy interactive mode.

Commands and Return Values

The following table lists the commands and return values.

<table>
<thead>
<tr>
<th>Command</th>
<th>Success Code</th>
<th>Return Value(s)</th>
<th>Asynchronous</th>
</tr>
</thead>
<tbody>
<tr>
<td>activate device</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>add advanced script</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>add authentication</td>
<td>200</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>add command script</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>add device</td>
<td>201</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>add device to group</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>Add diagnostic</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>add event</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>add group</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>Add group to parent group</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>Add parent group</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>add ip</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>add system message</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>add user</td>
<td>207</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>annotate access</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>annotate config</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>configure syslog</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>deactivate device</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>del access</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>del authentication</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>del device</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>del device data</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>del device from group</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>del drivers</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>del event</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>del group</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>Del group from parent group</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>del ip</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>Command</td>
<td>Success Code</td>
<td>Return Value(s)</td>
<td>Asynchronous</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------</td>
<td>-----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>del session</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>del script</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>del system message</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>del task</td>
<td>217</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>del user</td>
<td>211</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>deploy config</td>
<td>200</td>
<td>null</td>
<td>√</td>
</tr>
<tr>
<td>diff config</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>discover driver</td>
<td>200</td>
<td>null</td>
<td>√</td>
</tr>
<tr>
<td>discover drivers</td>
<td>200</td>
<td>null</td>
<td>√</td>
</tr>
<tr>
<td>get snapshot</td>
<td>200</td>
<td>null</td>
<td>√</td>
</tr>
<tr>
<td>list access</td>
<td>200</td>
<td>ResultSet</td>
<td></td>
</tr>
<tr>
<td>list access all</td>
<td>200</td>
<td>ResultSet</td>
<td></td>
</tr>
<tr>
<td>list basicip</td>
<td>200</td>
<td>Collection of String</td>
<td></td>
</tr>
<tr>
<td>list config</td>
<td>200</td>
<td>ResultSet</td>
<td></td>
</tr>
<tr>
<td>list config all</td>
<td>200</td>
<td>ResultSet</td>
<td></td>
</tr>
<tr>
<td>list device</td>
<td>501</td>
<td>ResultSet</td>
<td></td>
</tr>
<tr>
<td>list device data</td>
<td>200</td>
<td>ResultSet</td>
<td></td>
</tr>
<tr>
<td>list deviceinfo</td>
<td>200</td>
<td>Collection of String</td>
<td></td>
</tr>
<tr>
<td>list diagnostic</td>
<td>200</td>
<td>Collection of String</td>
<td></td>
</tr>
<tr>
<td>list drivers</td>
<td>200</td>
<td>ResultSet</td>
<td></td>
</tr>
<tr>
<td>list event</td>
<td>200</td>
<td>ResultSet</td>
<td></td>
</tr>
<tr>
<td>list groups</td>
<td>200</td>
<td>ResultSet</td>
<td></td>
</tr>
<tr>
<td>list icmp</td>
<td>200</td>
<td>Collection of String</td>
<td></td>
</tr>
<tr>
<td>list int</td>
<td>200</td>
<td>Collection of String</td>
<td></td>
</tr>
<tr>
<td>list ip</td>
<td>200</td>
<td>ResultSet</td>
<td></td>
</tr>
<tr>
<td>list ip all</td>
<td>200</td>
<td>ResultSet</td>
<td></td>
</tr>
<tr>
<td>list module</td>
<td>200</td>
<td>ResultSet</td>
<td></td>
</tr>
<tr>
<td>list ospfneighbor</td>
<td>200</td>
<td>Collection of String</td>
<td></td>
</tr>
<tr>
<td>list port</td>
<td>200</td>
<td>ResultSet</td>
<td></td>
</tr>
<tr>
<td>list routing</td>
<td>200</td>
<td>Collection of String</td>
<td></td>
</tr>
<tr>
<td>List script</td>
<td>200</td>
<td>ResultSet</td>
<td></td>
</tr>
<tr>
<td>list session</td>
<td>200</td>
<td>ResultSet</td>
<td></td>
</tr>
<tr>
<td>list system message</td>
<td>200</td>
<td>ResultSet</td>
<td></td>
</tr>
<tr>
<td>list task</td>
<td>200</td>
<td>ResultSet</td>
<td></td>
</tr>
<tr>
<td>Command</td>
<td>Success Code</td>
<td>Return Value (s)</td>
<td>Asynchronous</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>--------------</td>
<td>------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>list task all</td>
<td>513</td>
<td>ResultSet</td>
<td></td>
</tr>
<tr>
<td>list user</td>
<td>511</td>
<td>ResultSet</td>
<td></td>
</tr>
<tr>
<td>Mod advanced script</td>
<td>200</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>mod authentication</td>
<td>200</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Mod command script</td>
<td>200</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>mod device</td>
<td>204</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>Mod diagnostic</td>
<td>200</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>mod group</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>mod ip</td>
<td>200</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>mod module</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>mod port</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>mod task</td>
<td>215</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>mod unmanaged device</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>mod user</td>
<td>209</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>passwd</td>
<td>200</td>
<td>null</td>
<td>✓</td>
</tr>
<tr>
<td>pause polling</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>ping</td>
<td>200</td>
<td>String</td>
<td>✓</td>
</tr>
<tr>
<td>reload server options</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>resume polling</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>run advanced script</td>
<td>200</td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>run command script</td>
<td>200</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>run diagnostic</td>
<td>200</td>
<td>String</td>
<td>✓</td>
</tr>
<tr>
<td>run script</td>
<td>200</td>
<td>String</td>
<td>✓</td>
</tr>
<tr>
<td>show access</td>
<td>200</td>
<td>ResultSet</td>
<td></td>
</tr>
<tr>
<td>show basicip</td>
<td>200</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>show config</td>
<td>200</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>show device</td>
<td>200</td>
<td>ResultSet</td>
<td></td>
</tr>
<tr>
<td>show device config</td>
<td>200</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>show device latest diff</td>
<td>200</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>show deviceinfo</td>
<td>200</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>show diagnostic</td>
<td>200</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>show event</td>
<td>200</td>
<td>ResultSet</td>
<td></td>
</tr>
<tr>
<td>show fastlookup</td>
<td>200</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>show group</td>
<td>200</td>
<td>ResultSet</td>
<td></td>
</tr>
<tr>
<td>Command</td>
<td>Success Code</td>
<td>Return Value (s)</td>
<td>Asynchronous</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------</td>
<td>------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>show icmp</td>
<td>200</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>show int</td>
<td>200</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>show ip</td>
<td>200</td>
<td>ResultSet</td>
<td></td>
</tr>
<tr>
<td>show latest access</td>
<td>200</td>
<td>ResultSet</td>
<td></td>
</tr>
<tr>
<td>show module</td>
<td>200</td>
<td>ResultSet</td>
<td></td>
</tr>
<tr>
<td>show ospfneighbor</td>
<td>200</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>show polling status</td>
<td>200</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>show port</td>
<td>200</td>
<td>ResultSet</td>
<td></td>
</tr>
<tr>
<td>show routing</td>
<td>200</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>show script</td>
<td>200</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>show session</td>
<td>200</td>
<td>ResultSet</td>
<td></td>
</tr>
<tr>
<td>show session commands</td>
<td>200</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>show snapshot</td>
<td>200</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>show system message</td>
<td>200</td>
<td>ResultSet</td>
<td></td>
</tr>
<tr>
<td>show task</td>
<td>221</td>
<td>ResultSet</td>
<td></td>
</tr>
<tr>
<td>show user</td>
<td>219</td>
<td>ResultSet</td>
<td></td>
</tr>
<tr>
<td>synchronize</td>
<td>200</td>
<td>String [✓]</td>
<td></td>
</tr>
<tr>
<td>traceroute</td>
<td>200</td>
<td>String [✓]</td>
<td></td>
</tr>
</tbody>
</table>

**ResultSet Contents**

Where the Commands and Return Values table lists a `ResultSet` return type, these are the data types returned for columns 1 through N:

<table>
<thead>
<tr>
<th>Command</th>
<th>ResultSet Contents starting with column 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>list device data</td>
<td>java.lang.Integer deviceDataID</td>
</tr>
<tr>
<td>list conf</td>
<td>java.lang.String dataBlock</td>
</tr>
<tr>
<td>list conf all</td>
<td>java.lang.String blockType</td>
</tr>
<tr>
<td></td>
<td>java.util.Date createDate</td>
</tr>
<tr>
<td></td>
<td>java.lang.String comments</td>
</tr>
<tr>
<td></td>
<td>java.lang.Integer deviceAccessLogID</td>
</tr>
<tr>
<td></td>
<td>java.lang.Short blockFormat</td>
</tr>
<tr>
<td>list drivers</td>
<td>java.lang.Integer driverLookupID</td>
</tr>
<tr>
<td></td>
<td>java.lang.Integer deviceID</td>
</tr>
<tr>
<td></td>
<td>java.lang.String baseModelName</td>
</tr>
<tr>
<td></td>
<td>java.lang.String driverName</td>
</tr>
<tr>
<td>show access</td>
<td>java.lang.Integer deviceAccessLogID</td>
</tr>
<tr>
<td>list access</td>
<td>java.lang.String displayName</td>
</tr>
<tr>
<td>list access all</td>
<td>java.lang.String actionTaken</td>
</tr>
<tr>
<td>show latest access</td>
<td>java.lang.String accessTrigger</td>
</tr>
<tr>
<td></td>
<td>java.util.Date createDate</td>
</tr>
<tr>
<td></td>
<td>java.lang.Integer createUserID</td>
</tr>
<tr>
<td>Command</td>
<td>ResultSet Contents starting with column 1</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------------------------------</td>
</tr>
<tr>
<td>list device</td>
<td>java.lang.Integer deviceId</td>
</tr>
<tr>
<td>show device</td>
<td>java.lang.String primaryFQDN</td>
</tr>
<tr>
<td></td>
<td>java.lang.String hostName</td>
</tr>
<tr>
<td></td>
<td>java.lang.String primaryIPAddress</td>
</tr>
<tr>
<td></td>
<td>java.lang.String consoleIPAddress</td>
</tr>
<tr>
<td></td>
<td>java.lang.String nATIPAddress</td>
</tr>
<tr>
<td></td>
<td>java.lang.String tFTPServerIPAddress</td>
</tr>
<tr>
<td></td>
<td>java.lang.Integer consolePort</td>
</tr>
<tr>
<td></td>
<td>java.lang.String deviceName</td>
</tr>
<tr>
<td></td>
<td>java.lang.String serialNumber</td>
</tr>
<tr>
<td></td>
<td>java.lang.String assetTag</td>
</tr>
<tr>
<td></td>
<td>java.lang.String softwareVersion</td>
</tr>
<tr>
<td></td>
<td>java.lang.String firmwareVersion</td>
</tr>
<tr>
<td></td>
<td>java.lang.String vendor</td>
</tr>
<tr>
<td></td>
<td>java.lang.String model</td>
</tr>
<tr>
<td></td>
<td>java.lang.String deviceType</td>
</tr>
<tr>
<td></td>
<td>java.lang.String geographicalLocation</td>
</tr>
<tr>
<td></td>
<td>java.lang.String timeZone</td>
</tr>
<tr>
<td></td>
<td>java.lang.String deviceFunction</td>
</tr>
<tr>
<td></td>
<td>java.lang.String comments</td>
</tr>
<tr>
<td></td>
<td>java.util.Date createDate</td>
</tr>
<tr>
<td></td>
<td>java.util.Date lastAccessAttemptDate</td>
</tr>
<tr>
<td></td>
<td>java.util.Date lastAccessSuccessDate</td>
</tr>
<tr>
<td></td>
<td>java.util.Date lastSnapshotDate</td>
</tr>
<tr>
<td></td>
<td>java.lang.String lastAccessAttemptStatus</td>
</tr>
<tr>
<td></td>
<td>java.lang.Integer lastModifiedUserID</td>
</tr>
<tr>
<td></td>
<td>java.lang.Short excludeFromPoll</td>
</tr>
<tr>
<td></td>
<td>java.lang.Short canUseChangeAgents</td>
</tr>
<tr>
<td></td>
<td>java.lang.String accessMethods</td>
</tr>
<tr>
<td></td>
<td>java.lang.String modemNumber</td>
</tr>
<tr>
<td></td>
<td>java.lang.Short managementStatus</td>
</tr>
<tr>
<td></td>
<td>java.lang.String feedSource</td>
</tr>
<tr>
<td></td>
<td>java.util.Date lastImportDate</td>
</tr>
<tr>
<td></td>
<td>java.util.Date lastRecordModifiedDate</td>
</tr>
<tr>
<td></td>
<td>java.lang.String changeEventData</td>
</tr>
<tr>
<td></td>
<td>java.lang.Integer mostRecentConfigID</td>
</tr>
<tr>
<td></td>
<td>java.lang.Integer lastConfigChangeUserID</td>
</tr>
<tr>
<td>Command</td>
<td>ResultSet Contents starting with column 1</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------------------------</td>
</tr>
</tbody>
</table>
| list groups | java.lang.Integer deviceGroupID  
java.lang.String deviceGroupName  
java.util.Date createDate  
java.lang.String comments  
java.lang.String deviceGroupCustom1  
java.lang.String deviceGroupCustom2  
java.lang.String deviceGroupCustom3  
java.lang.String deviceGroupCustom4  
java.lang.String deviceGroupCustom5  
java.lang.String deviceGroupCustom6  
java.lang.Integer deviceCount |
| list session | java.lang.Integer interceptorLogID  
java.util.Date startDate  
java.util.Date endDate  
java.lang.Integer userID  
java.lang.Integer deviceID  
java.lang.String deviceIP  
java.lang.String sessionType  
java.lang.String sessionData  
java.lang.Short status  
java.lang.String interceptorLogCustom1  
java.lang.String interceptorLogCustom2  
java.lang.String interceptorLogCustom3  
java.lang.String interceptorLogCustom4  
java.lang.String interceptorLogCustom5  
java.lang.String interceptorLogCustom6 |
| list system message | java.lang.Integer eventID  
java.lang.Integer eventUserID  
java.lang.Integer eventDeviceID  
java.lang.String eventType  
java.util.Date eventDate  
java.lang.Short eventClass  
java.lang.Integer eventTaskID  
java.lang.String eventText |
| list task | java.lang.Integer scheduleTaskID  
java.lang.Integer deviceGroupID  
java.lang.Integer succeededChildCount  
java.lang.Integer failedChildCount  
java.lang.Integer pendingChildCount  
java.lang.Integer parentTaskID |
<table>
<thead>
<tr>
<th>Command</th>
<th>ResultSet Contents starting with column 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>list user</td>
<td>java.lang.Integer userID</td>
</tr>
<tr>
<td>show user</td>
<td>java.lang.String username</td>
</tr>
<tr>
<td></td>
<td>java.lang.String firstName</td>
</tr>
<tr>
<td></td>
<td>java.lang.String lastName</td>
</tr>
<tr>
<td></td>
<td>java.lang.String userPassword</td>
</tr>
<tr>
<td></td>
<td>java.lang.String emailAddress</td>
</tr>
<tr>
<td></td>
<td>java.util.Date createDate</td>
</tr>
<tr>
<td></td>
<td>java.lang.String timeZone</td>
</tr>
<tr>
<td></td>
<td>java.lang.Short requiredUser</td>
</tr>
<tr>
<td></td>
<td>java.lang.String aaaUserName</td>
</tr>
<tr>
<td></td>
<td>java.lang.String aaaPassword</td>
</tr>
<tr>
<td></td>
<td>java.lang.Short useAaaLoginForProxy</td>
</tr>
<tr>
<td></td>
<td>java.lang.String userCustom1</td>
</tr>
<tr>
<td></td>
<td>java.lang.String userCustom2</td>
</tr>
<tr>
<td></td>
<td>java.lang.String userCustom3</td>
</tr>
<tr>
<td></td>
<td>java.lang.String userCustom4</td>
</tr>
<tr>
<td></td>
<td>java.lang.String userCustom5</td>
</tr>
<tr>
<td></td>
<td>java.lang.String userCustom6</td>
</tr>
<tr>
<td>show event</td>
<td>java.lang.Integer eventID</td>
</tr>
<tr>
<td>list event</td>
<td>java.lang.Integer eventUserID</td>
</tr>
<tr>
<td></td>
<td>java.lang.Integer eventDeviceID</td>
</tr>
<tr>
<td></td>
<td>java.lang.String eventType</td>
</tr>
<tr>
<td></td>
<td>java.util.Date eventDate</td>
</tr>
<tr>
<td></td>
<td>java.lang.Short eventClass</td>
</tr>
<tr>
<td></td>
<td>java.lang.Integer eventTaskID</td>
</tr>
<tr>
<td></td>
<td>java.lang.String eventText</td>
</tr>
<tr>
<td></td>
<td>java.lang.String eventData</td>
</tr>
<tr>
<td></td>
<td>java.lang.Integer configPolicyID</td>
</tr>
<tr>
<td>show ip</td>
<td>java.lang.Integer ipAddress</td>
</tr>
</tbody>
</table>

PERL API Reference Guide 23
<table>
<thead>
<tr>
<th>Command</th>
<th>ResultSet Contents starting with column 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>list ip</td>
<td>java.lang.String ipValue</td>
</tr>
<tr>
<td>list ip all</td>
<td>java.lang.String ipMask</td>
</tr>
<tr>
<td></td>
<td>java.lang.Integer ipPriority</td>
</tr>
<tr>
<td></td>
<td>java.lang.String ipName</td>
</tr>
<tr>
<td></td>
<td>java.lang.String comments</td>
</tr>
<tr>
<td></td>
<td>java.util.Date changeDate</td>
</tr>
<tr>
<td></td>
<td>java.lang.Short ipType</td>
</tr>
<tr>
<td></td>
<td>java.lang.Short usedToAccess</td>
</tr>
<tr>
<td></td>
<td>java.lang.Integer devicePortID</td>
</tr>
<tr>
<td></td>
<td>java.lang.Integer lastModifiedUserID</td>
</tr>
<tr>
<td></td>
<td>java.lang.Integer deviceID</td>
</tr>
<tr>
<td>show module</td>
<td>Integer deviceModuleID</td>
</tr>
<tr>
<td>list module</td>
<td>Integer deviceID</td>
</tr>
<tr>
<td></td>
<td>String slot</td>
</tr>
<tr>
<td></td>
<td>String moduleModel</td>
</tr>
<tr>
<td></td>
<td>String moduleDescription</td>
</tr>
<tr>
<td></td>
<td>String moduleOS</td>
</tr>
<tr>
<td></td>
<td>String firmwareVersion</td>
</tr>
<tr>
<td></td>
<td>String hardwareRevision</td>
</tr>
<tr>
<td></td>
<td>Integer memory</td>
</tr>
<tr>
<td></td>
<td>String moduleCustom1</td>
</tr>
<tr>
<td></td>
<td>String moduleCustom2</td>
</tr>
<tr>
<td></td>
<td>String moduleCustom3</td>
</tr>
<tr>
<td></td>
<td>String moduleCustom4</td>
</tr>
<tr>
<td></td>
<td>String moduleCustom5</td>
</tr>
<tr>
<td></td>
<td>String moduleCustom6</td>
</tr>
<tr>
<td></td>
<td>String comments</td>
</tr>
<tr>
<td></td>
<td>String serialNumber</td>
</tr>
<tr>
<td>show port</td>
<td>Integer devicePortID</td>
</tr>
<tr>
<td>list port</td>
<td>Integer deviceID</td>
</tr>
<tr>
<td></td>
<td>String portCustom1</td>
</tr>
<tr>
<td></td>
<td>String portCustom2</td>
</tr>
<tr>
<td></td>
<td>String portCustom3</td>
</tr>
<tr>
<td></td>
<td>String portCustom4</td>
</tr>
<tr>
<td></td>
<td>String portCustom5</td>
</tr>
<tr>
<td></td>
<td>String portCustom6</td>
</tr>
<tr>
<td></td>
<td>String comments</td>
</tr>
<tr>
<td></td>
<td>String portName</td>
</tr>
<tr>
<td></td>
<td>String portAllows</td>
</tr>
<tr>
<td></td>
<td>String portType</td>
</tr>
<tr>
<td></td>
<td>String portStatus</td>
</tr>
<tr>
<td></td>
<td>String description</td>
</tr>
</tbody>
</table>
Appendix A: NCM Documentation

To open any of the available documents, after logging into NCM, on the menu bar click Docs. The CiscoWorks Network Compliance Manager Documentation page opens. Click the title of the document you want to view in PDF. NCM also provides context-sensitive help that you can access via the Help icon on the top of each page of the Web interface.

All documentation, including this document and any or all of the parts of the NCM documentation set, might be upgraded over time. Therefore, we recommend you access the NCM documentation set using the Cisco.com URL:


The Docs tab visible from within Network Compliance Manager might not include links to the latest documents.

- *User Guide for Network Compliance Manager 1.3* — Includes information on how to use NCM.
- Context-Sensitive Help — Click the Help icon on any page for Help.
- *Device Driver Reference for Network Compliance Manager 1.3* — Includes device-specific information for configuring devices to work with NCM.
- *PERL, Java, and SOAP API Reference Guides* — Includes instructions for using the Application Programming Interfaces for PERL, Java, and SOAP.
Appendix B: Obtaining Documentation, Obtaining Support, and Security Guidelines

For information on obtaining documentation, obtaining support, providing documentation feedback, security guidelines, and also recommended aliases and general Cisco documents, see the monthly What's New in Cisco Product Documentation, which also lists all new and revised Cisco technical documentation, at:

activate device
Mark a device as activated.

Synopsis
activate device [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>]

Description
- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -host - A valid hostname
- -fqdn - A valid Fully Qualified Domain Name
- -deviceid - A device ID

Examples
- activate device -ip 192.0.2.10
- activate device -ip "East Site:192.0.2.10"

add advanced script
Add a new advanced script.

Synopsis
add advanced script -name <Name> [-description <Description>] [-scripttype <Script Type>] [-family <Device Family>] -language <Script Language> [-parameters <Parameters>] -script <Script Text>

Description
- -name - Name for the new advanced script
- -description - Description for the new advanced script
- -scripttype - Script type (i.e. user defined subcategory)
- -family - Device family for the new advanced script
- -language - Language for the new advanced script - must be a supported language such as Expect or Perl
- -parameters - Command line parameters for the new advanced script
- -script - Script text
Examples

- add advanced script -name "Extended Ping" -description "Run extended ping to desired address" -scripttype "Troubleshooting scripts" -family "Cisco IOS" -language "Expect" -parameters "-l /usr/etc/log.txt" -script "send \"extended ping $Target_IP\""

add authentication

Modify device password information.

Synopsis


Description

This command can modify passwords on a specific device or device group, or merely update what the system knows of a device's or network's password information. The -ip option provides information specific to the device. Otherwise, the command adds a network-wide password rule to the system. When using this command to modify passwords on a device, the modification operation is actually a scheduled task.

- -loc - The location to which password information should be written. Valid values for this argument are "db", "device", and "group". "db" tells the command that password information should be changed only in the system's database. "device" tells the command that the password changes should be made on the device as well and "group" performs the same function as "device" but across all devices in the group.
- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.: The device to which this password information should apply.
- -host - A valid hostname: An existing device to which this password information should apply.
- -fqdn - A valid Fully Qualified Domain Name: An existing device to which this password information should apply.
- -deviceid - A device ID
-snmpro - When used in conjunction with -loc db, this argument is taken as a single community string understood by the system as THE read only community string for the device or network. When used in conjunction with -loc device, this argument is taken as a comma-separated list of read only community strings to be, either set on the device, or appended to an existing list of read only community strings (depends on whether or not the -appendsnmpro flag was supplied.)

-snmprw - When used in conjunction with -loc db, this argument is taken as a single community string understood by the system as THE read write community string for the device or network. When used in conjunction with -loc device, this argument is taken as a comma-separated list of read write community strings to be, either set on the device, or appended to an existing list of read write community strings (depends on whether or not the -appendsnmprw flag was supplied.)

-snmpv3user - When used in conjunction with -loc db, this argument is taken as the username for snmpv3 access.

-snmpv3authpw - When used in conjunction with -loc db, this argument is taken as the authentication password for snmpv3 access.

-snmpv3encryptpw - When used in conjunction with -loc db, this argument is taken as the encryption password for snmpv3 access.

-user - Username.

-passwd - Password.

-enableuser - ADDITIONAL username to get to "enable" mode.

-enablepasswd - ADDITIONAL password to get to "enable" mode.

-connectionmethods - The methods used by the system to connect to devices. Can be telnet, serial, direct, or SSH.

-accessvariables - To override variables in the script, such as prompts.

-start - YYYY:MM:DD:HH:mm. The first date on which the task will run. Use this option only if the argument to the -loc flag is "device".

-appendsnmpro - Supply this option if read only community strings should be appended to any existing on the device. Use this option only if the argument to the -loc flag is "device".

-appendsnmprw - Supply this option if read write community strings should be appended to any existing on the device. Use this option only if the argument to the -loc flag is "device".

-sync - Indicates that the command should return only after the password change task is complete. Do not use this option with -start.

-group - The group name for performing this command across all devices in a group.

Examples

- add authentication -loc db -ip 192.0.2.10 -passwd fish -snmpro public -enablepasswd 31337
- add authentication -loc db -ip 192.0.2.10 -passwd old -enablepasswd joshua -snmpro public -snmprw public
- add authentication -loc device -ip 192.0.2.10 -passwd limited -enablepasswd full
- add authentication -loc device -ip 192.0.2.10 -passwd some -enablepasswd all -snmprw brillig,slithy,toves,gire -appendsnmprw -sync
- add authentication -loc device -ip 192.0.2.10 -passwd less -enablepasswd more -snmpro foo,bar,fork,snork -start 2004:02:29:23:59
add authentication -loc group -group MyDevices -passwd less -enablepasswd more -snmp pro foo, bar, fork, snork -start 2004:02:29:23:59

---

add command script
Add a new command script.

Synopsis
add command script -name <Name> [-description <Description>] [-scripttype <Script Type>] -mode <Mode> [-driver <Driver List>] -script <Script Text>

Description
- -name - Name for the new command script
- -description - Description for the new command script
- -scripttype - Script type (i.e. user defined subcategory)
- -mode - Command script mode
- -driver - List of applicable drivers - provided as a comma separated list of internal driver names
- -script - Script text

Examples
- add command script -name "Extended Ping" -description "Run extended ping to desired address" -scripttype "Troubleshooting scripts" -mode "Cisco IOS enable" -driver "CiscoIOSGeneric,CiscoIOSSwitch" -script "extended ping $Target_IP$"

---

add device
Add a device to the system.

Synopsis

Description
- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device will be put in.
- -hostname - The device’s host name
- -comment - Additional information regarding the device.
- -description - The descriptive name of the device (informational only).
- -model - The device's model (such as 2620).
- vendor - The device's vendor (such as Cisco).
- domain - A fully qualified domain name (such as www.google.com).
- serial - The device's serial number.
- asset - The device's asset tag.
- location - The device's location.
- unmanaged - 0: Mark this device as managed by the system. 1: Mark this device to be unmanaged by the system.
- nopoll - 0: Mark this device to be polled for changes. 1: Mark this device as not to be polled for changes.
- consoleip - a.b.c.d where 0 <= a,b,c,d <= 255
- consoleport - The port number
- tftpserverip - a.b.c.d where 0 <= a,b,c,d <= 255
- natip - a.b.c.d where 0 <= a,b,c,d <= 255
- useconsoleserver - true, if the device uses a console server. false, if the device does not. If this option is not provided, it is assumed that the device does not use a console server.
- accessmethods - A comma-separated list of access methods, or "none". The set of access methods: {telnet, ssh, rlogin, SCP, FTP, TFTP, SNMP, snmp_noauthnopriv, snmp_authnopriv, snmp_authpriv}. If this option is not provided, the system will try all access methods when attempting to connect to the device.
- hierarchylayer - This device attribute is used in diagramming. When you config a network diagram, you can select which hierarchy layers on which to filter. Valid values include: (core, distribution, access, edge and "layer not set").

Examples

- add device -ip 192.0.2.10
- add device -ip "East Site:192.0.2.10"
- add device -ip 192.0.2.10 -model 3460 -vendor Cisco
- add device -ip 192.0.2.10 -comment "the web server." -domain www.minibosses.com
- add device -ip 192.0.2.10 -consoleip 192.0.2.10 -consoleport 62888 -useconsoleserver true -accessmethods ssh,SNMP

---

**add device to group**

Add a device to a device group.

**Synopsis**

add device to group [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] -group <Device group>

**Description**

- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -host - A valid hostname
- -fqdn - A valid Fully Qualified Domain Name
- -deviceid - A device ID
-group - The name of the device group to which the device should be added.

Examples

- add device to group -ip 192.0.2.10 -group tech-dev
- add device to group -ip "Default Site:192.0.2.10" -group tech-dev

add diagnostic

Add a new custom diagnostic script.

Synopsis

add diagnostic -name <Name> [-description <Description>] -mode <Mode> [-driver <Driver List>] -script <Script Text>

Description

- -name - Name for the new diagnostic
- -description - Description for the new diagnostic
- -mode - Command script mode
- -driver - List of applicable drivers - provided as a comma separated list of internal driver names
- -script - Diagnostic script text

Examples

- add diagnostic -name "Show IP CEF" -description "Gather IP CEF information" -mode "Cisco IOS enable" -driver "CiscoIOSGeneric,CiscoIOSSwitch" -script "show ip cef"

add event

Add an event.

Synopsis

add event -message <Event> [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>]

Description

An email message (containing the event) will be the result of an added events if the system is configured to send email for added events.

- -message - The text of the event
- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -host - A valid hostname
- -fqdn - A valid Fully Qualified Domain Name
Examples

- add event -ip 192.0.2.10 -message "Connectivity to the border router has been restored."
- add event -message "This is a test of the emergency broadcast system."

---

add event rule
Add a event rule.

Synopsis
add event rule -name <Event Rule Name> -action <Event Action> -receiverhost <Hostname or IP Address> [-receiverport <Port>] [-events <List of Event Types>] [-community <Community String>]

Description
Add new event rule. It will subscribe provided host to the system events.

- -name - The name identifier for event rule
- -action - event type, for now only snmp supports, use -action snmp
- -receiverhost - A valid hostname or ip address
- -receiverport - A numeric port, if not provided, then 162 will be used
- -events - List of event types, separated by column. If not provided, then ALL will be used
- -community - Community string, if not provided, then public will be used

Examples

- add event rule -name Name1 -receiverhost host1 -action snmp -community private -events "Device Added:Device Deleted"
- add event rule -name Name2 -receiverhost host2 -action snmp

---

add group
Add a group to the system.

Synopsis
add group -name <Name> -type <Type> [-comment <Comment>] [-shared <Shared>]

Description

- -name - The name of the group to add.
- -type - The type of the group to add. "device" and "user" are the valid values for this option.
- -comment - Additional information about the group.
- -shared - 1 if the group is shared, 0 if it is not.
Examples

- add group -name "border routers" -type device -comment "The group containing all border routers."

add group to parent group
Add a device group to a parent device group.

Synopsis
add group to parent group -parent <Parent group name> -child <Child group name>

Description

- -parent - Name of the parent group
- -child - Name of the child group

Examples

- add group to parent group -parent "North America" -child "West Region"

add image
Add images to database.

Synopsis
add image -imageset <imageset name> -images <images> [-driver <driver name>] [-model <model name>] [-memory <minimum system memory (in bytes)>] [-processor <processor name>] [-bootrom <BootROM name>]

Description

Add images to database. Must specify either driver or model

- -imageset - The imageset the images will add to.
- -images - The images to add. The paths specified by this option must point to files accessible by the management server. Files must be placed on the management server first.
- -driver - The driver the images required.
- -model - The device model the images required.
- -memory - The minimum system memory required (in bytes) for images.
- -processor - The hardware required for images.
- -bootrom - The BootROM required for images.

Examples

- add image -imageset fooiset -images c:\data\bar.bin -driver CiscoPIX
- add image -imageset fooiset -images c:\data\bar.bin,c:\images\foobar.bin -model "WS-C2924M-XL-EN (C2900XL series)"
add imageoption
Add information for device which is not under management but needed for software image update.

Synopsis
add imageoption [-imagemodels <device model names (separated by ','),>] [-imageprocessors <device processor names (separated by ','),>] [-imagebootroms <device bootROM names (separated by ','),>]

Description
Add information for device which is not under management but needed for software image update.

- -imagemodels - device model to be added.
- -imageprocessors - device processors to be added.
- -imagebootroms - device BootROMs to be added.

Examples
- add imageoption -imagemodels model1,model2 -imageprocessors PP3,PP4

add ip
Add new secondary ip.

Synopsis

Description
- -ipvalue - The ip value a.b.c.d where 0 <= a,b,c,d <= 255
- -deviceip - The device's ip address a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -comment - Additional information regarding the device.
- -usetoaccess - Use this IP Value to access its device, 0 - yes, 1 - no, default - no
- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -host - A valid hostname
- -fqdn - A valid Fully Qualified Domain Name
- -deviceid - A device ID
Examples

- add ip -deviceip 192.0.2.10 -ipvalue 192.0.2.10 -comment "my own ip"
- add ip -deviceip 192.0.2.10 -ipvalue 192.0.2.10 -usetoaccess 0
- add ip -deviceid 1401 -ipvalue 192.0.2.10 -usetoaccess 0

---

add metadata

Add a piece of custom data to be associated with a specific field and associated entity.

**Synopsis**

add metadata -fieldid <Metadata Field ID> [-data <Data>] -associatedtableid <Matching Row ID>

**Description**

- **-fieldid** - Field ID the data is to be associated with
- **-data** - Data to be associated, if not included, data is null
- **-associatedtableid** - ID of the associated row the data coresponds to

**Examples**

- add metadata -fieldid 121 -associatedtableid 21031
- add metadata -fieldid 121 -data Room101 -associatedtableid 21031

---

add metadata field

Used to define a custom data field for a specific table.

**Synopsis**

add metadata field -fieldname <Field Name> [-fieldvalues <Field Values>] [-inuse <In Use>] [-flags <Allow HTML>] -associatedtable <Associated Table>

**Description**

- **-fieldname** - Name of the field to be added
- **-fieldvalues** - List of comma separated values that the field is restricted to. If not specified, the value for this field is not restricted
- **-inuse** - Turns the field on or off. 1 is on, 0 is off. When the field is off, it will not be displayed with the other custom fields.
- **-flags** - Used for allowing HTML in the field value. 1 is allow, 0 is disallow. If disallowed, HTML will be escaped for displaying.
- **-associatedtable** - The table to associate this field with

**Examples**

- add metadata field -fieldname Room -fieldvalues 101,102,103,104 -inuse 1 -flags 0 -associatedtable RN_DEVICE
• add metadata field -fieldname Building -inuse 1 -flags 0 -associatedtable RN_DEVICE

---

**add parent group**
Add a parent group to the system.

**Synopsis**
add parent group -name <Name> -type <Type> [-comment <Comment>]

**Description**
- **-name** - The name of the parent group to add.
- **-type** - The type of the parent group to add. "device" is currently the only valid argument to this option.
- **-comment** - Additional information about the parent group.

**Examples**
- add parent group -name "North America" -type device -comment "Parent group to roll up East, Central and West regions."

---

**add partition**
Add a partition to view.

**Synopsis**
add partition -viewname <Viewname> -name <Name> [-comment <Comment>]

**Description**
- **-viewname** - The name of the view this partition goes to.
- **-name** - The name of the partition to add.
- **-comment** - Additional information about the partition.

**Examples**
- add partition -viewname "Site" -name Redmond -comment "Redmond Site"

---

**add system message**
Add a system message.

**Synopsis**
add system message -message <System Message> [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>]
Description
An email message (containing the system message) will be the result of an added system messages if the system is configured to send email for added events.

- message - The text of the system message
- ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- host - A valid hostname
- fqdn - A valid Fully Qualified Domain Name
- deviceid - A device ID

Examples
• add system message -ip 192.0.2.10 -message "Connectivity to the border router has been restored."
• add system message -message "This is a test of the emergency broadcast system."

add user
Add a user to the system.

Synopsis
add user -u <Username> -p <Password> -fn <First name> -ln <Last name> [-email <Email address>] [-aaausername <Username>] [-aaapassword <AAA Password>] [-useaaaloginforproxy <Use AAA Logins for Proxy (yes|no)>] [-extauthfailover <Allow External Auth Failover (yes|no)>]

Description
• -u - Username
• -p - Password
• -fn - First name
• -ln - Last name
• -email - Email address
• -aaausername - AAA username for this user.
• -aaapassword - AAA password for this user.
• -useaaaloginforproxy - Whether to user AAA logins for the Proxy Interface for this user (yes|no).
• -extauthfailover - Whether to allow external auth failover for this user (yes|no).

Examples
• add user -u johnd -p fish -fn john -ln doe -email johnd@example.net

add user to group
Add a user to a user group.
Synopsis
add user to group -u <Username> -g <User group name>

Description
• -u - Username
• -g - User group name

Examples
• add user to group -u johnd -g "User Group 1"

annotate access
Modify the comments on, or the display name of, a device access record.

Synopsis
annotate access -id <Device access record ID> [-comment <Comment>] [-name <Name>] [-customname <Custom name>] [-customvalue <Custom value>]

Description
• -id - Specifies a device access record. Think of this as a "device access record ID".
• -comment - Additional information regarding the access record.
• -name - An optional name for the access record.
• -customname - The custom field name
• -customvalue - The custom field value

Examples
• annotate access -id 2 -comment "Device tainted at this point." -name "Intrusion detected"
• annotate access -id 2 -customname TicketID - customvalue 5

annotate config
Add a comment to the specified config.

Synopsis
annotate config -id <Config ID> -comment <comment>

Description
Note that comments added by means of this command are not added to the config itself. They are stored separately along with the config.

• -id - The ID of the config on which you are commenting.
• -comment - Additional information regarding the config.
Examples

- annotate config -id 1754 -comment "north campus group template."

assign driver
Manually assign driver to device.

Synopsis
assign driver [-ip <IP address>] [-id <Device ID>] -name <Driver Name>

Description

- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -id - A valid device id
- -name - A valid internal driver name, supported by system

Examples

- assign driver -ip 192.0.2.10 -name CiscoIOSGenericNoLog
- assign driver -id 70 -name CiscoIOSGenericNoLog

configure syslog
Configure a device to send syslog messages to the system's change detection facilities.

Synopsis

Description
Have the system configure the specified device to send all syslog messages necessary for the system's change detection facilities to function optimally to the system's syslog server. The configuration operation is actually a scheduled task.

- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -group - A valid group name. Do not use this option with -ip (exactly one of -ip or -group must be specified).
- -host - A valid hostname
- -fqdn - A valid Fully Qualified Domain Name
- -deviceid - A device ID
- -rep - (#min | #:#: | #days | #weeks | #months) where # is a positive integer. #:# is hours:minutes—the two integers do not have to be the same. Do not use this option with -sync.
• -sync - Indicates the command should return only after the Configure Syslog task is complete. Do not use this option with -rep or -start.
• -start - YYYY:MM:DD:HH:mm. The first date on which the task will run.
• -comment - An optional comment about the Configure Syslog task.
• -usesyslogrelay - Indicates to the syslog configuration task that the device currently logs to syslog relay host. Supply this option if you wish to set up forwarding on that relay host rather than have the device log directly to the system's syslog server. The specified IP address is taken to be the IP address of the relay host.

Examples

- configure syslog -ip 192.0.2.10
- configure syslog -ip 192.0.2.10 -usesyslogrelay blanka
- configure syslog -ip 192.0.2.10 -sync
- configure syslog -group mygroup

connect
Connect to a device.

Synopsis


Description

Connect to a device through the system's Proxy Interface via telnet, ssh, or rlogin. If you are connected to a device through a console server, you may hit ctrl-\ to return to the system shell after logging out of the device.

• -login - Bypass single sign-on and instead take the user to the device login prompt.
• -method - Method used to connect to devices outside of the system or for devices in the system when single sign-on is turned off (implies -login option).
• -override - Force a connection to a device in the event that simultaneous connection warning or prevention is turned on.
• -info - Dump connection variable information (can set the info prefix following a colon, like "-info:"")
• -ignoreptyerrors - Ignore pty errors for SSHv2 connections if "-login" option is on.
• - Hostname, Device ID, Fully Qualified Domain Name, or Primary IP Address to use to lookup the device to connect to. The characters * and ? can be used as wildcards. The device id can be specified instead by preceding it with a '#'
• - Port to use to connect to devices outside of the system.

Examples

• connect 192.0.2.10
• connect -login Zangief
connect -override mydevice

deactivate device
Mark a device as deactivated.

Synopsis
deactivate device [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>]

Description
- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -host - A valid hostname
- -fqdn - A valid Fully Qualified Domain Name
- -deviceid - A device ID

Examples
- deactivate device -host rtr5.vfm.lab

del access
Delete access records.

Synopsis
del access [-id <Device Access Record ID.>] [-cutoff <Date>]

Description
This command can delete a single access record when provided that record's id (via. the option "-id"), or all access records prior to a given date (via the option "-cutoff"). Provide exactly one of "-id", "-cutoff". Note that deleting access records will cause all configs associated with the deleted access record to also be deleted.

- -id - A device access record ID.
- -cutoff - YYYY:MM:DD:HH:mm. All access records prior to this date will be deleted.

Examples
- del access -id 6288
- del access -cutoff 2004:02:29:23:59
**del authentication**

Deletes all password information associated with the specified device.

**Synopsis**

del authentication [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>]

**Description**

- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in. The device for which password information should be deleted.
- -host - A valid hostname: The device for which password information should be deleted.
- -fqdn - A valid Fully Qualified Domain Name: The device for which password information should be deleted.
- -deviceid - A device ID

**Examples**

- del authentication -ip 192.0.2.10

---

**del device**

Delete the specified device.

**Synopsis**

del device [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>]

**Description**

- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -host - A valid hostname
- -fqdn - A valid Fully Qualified Domain Name
- -deviceid - A device ID

**Examples**

- del device -ip 192.0.2.10
- del device -ip "East Site:192.0.2.10"
del device data
Delete device configuration and diagnostic data.

Synopsis
del device data [-id <Config ID>] [-cutoff <Date>]

Description
This command can delete a single device data block when provided that device data id (via the option "-id"), or all device data prior to a given date (via the option "-cutoff"). Provide exactly one of "-id", "-cutoff".

- -id - A config ID
- -cutoff - YYYY:MM:DD:HH:mm. All configs prior to this date will be deleted.

Examples
- del device data -id 866227436

del device from group
Delete a device from a device group.

Synopsis
del device from group [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] -group <Device group>

Description
- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -host - A valid hostname
- -fqdn - A valid Fully Qualified Domain Name
- -deviceid - A device ID
- -group - The name of the device group from which the device should be deleted.

Examples
- del device from group -ip 192.0.2.10 -group tech-dev

del drivers
Delete all drivers associated with a device.

Synopsis
del drivers [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>]

44 PERL API Reference Guide
Description

- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -host - A valid hostname
- -fqdn - A valid Fully Qualified Domain Name
- -deviceid - A device ID

Examples

- del drivers -ip 192.0.2.10

---

del event
Delete the specified event.

Synopsis

del event -id <event ID>

Description

- -id - A valid event id

Examples

- del event -id 799

---

del group
Delete a group from the system.

Synopsis

del group -name <Name> -type <Type>

Description

Specify the group by both its name and type.

- -name - The name of the group to be removed.
- -type - The type of the group to be removed.

Examples

- del group -name "border routers" -type "device"

---

del group from parent group
Remove a device group from a parent device group.
**Synopsis**

del group from parent group -parent <Parent group name> -child <Child group name>

**Description**

- -parent - Name of the parent group  
- -child - Name of the child group

**Examples**

- del group from parent group -parent "North America" -child "Costa Rica NOC"

---

**del ip**

Delete the specified ip.

**Synopsis**

del ip -ipvalue <Value> [-deviceip <Device IP address>] [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>]

**Description**

- -ipvalue - The ip value a.b.c.d where 0 <= a,b,c,d <= 255  
- -deviceip - The device's ip address a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.  
- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.  
- -host - A valid hostname  
- -fqdn - A valid Fully Qualified Domain Name  
- -deviceid - A device ID

**Examples**

- del ip -deviceip 192.0.2.10 -ipvalue 192.0.2.10  
- del ip -deviceid 1401 -ipvalue 192.0.2.10

---

**del metadata**

Delete a specific piece of custom data.

**Synopsis**

del metadata -metadataid <Metadata ID>

**Description**

- -metadataid - ID of the custom data to delete
Examples

- del metadata -metadatalid 54535

---

del metadata field
Delete a custom data field and all data currently associated with that field.

Synopsis
del metadata field -fieldid <Field ID>

Description

- -fieldid - ID of the custom data field to delete

Examples

- del metadata field -fieldid 8394

---

del partition
Delete a partition from view.

Synopsis
del partition -name <Name>

Description

- -name - The name of the partition to be removed.

Examples

- del partition -name "Redmond Site"

---

del script
Delete an existing command script, advanced script or diagnostic.

Synopsis
del script [-id <Script / Diagnostic ID>] [-name <Script / Diagnostic Name>] [-type <Script / Diagnostic Type>]

Description
Delete the indicated command script, advanced script or diagnostic. The desired script or diagnostic can be specified by ID, or by a combination of name and type. If more than one name match occurs, then an error will be reported and you must specify the unique script desired by ID.
• -id - ID of the desired script or diagnostic
• -name - Name of the desired script or diagnostic
• -type - Type of the desired script or diagnostic - may be command, advanced or diagnostic

Examples

• del script -id 5
• del script -name "Edit Port Duplex" -type command

del session
Delete an interceptor log record.

Synopsis
del session -id <Interceptor log id>

Description

• -id - Interceptor log ID

Examples

• del session -id 5

del system message
Delete the specified system message.

Synopsis
del system message -id <System message ID>

Description

• -id - A valid system message id

Examples

• del system message -id 799

del task
Delete a task.

Synopsis
del task -id <Task ID>
**Description**
Deletes a task, whether it has run or not.

- **-id** - A task ID

**Examples**

- del task-id 4321

---

**del user**
Delete a user from the system.

**Synopsis**

```
del user -u <User name>
```

**Description**

- **-u** - The user name to be deleted

**Examples**

- del user -u johnd

---

**del user from group**
Delete a user from a user group.

**Synopsis**

```
del user from group -u <Username> -g <User group name>
```

**Description**

- **-u** - Username
- **-g** - User group name

**Examples**

- del user from group -u johnd -g "User Group 1"

---

**delete image**
Delete software images from database

**Synopsis**

```
delete image -imageset <imageset name> -images <images separated by ,>
```
Description
Delete software images from database

- `imageset` - imageset name the images will be deleted.
- `images` - images to be deleted.

Examples

- `delete image -imageset fooset -images bar.bin,baz.bin`

---

deploy config

Deploy config to a device.

Synopsis

Description
Deploy the specified config to a specified device either right away, or at some point in the future. The deploy operation is actually a scheduled task.

- `ip` - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- `host` - A valid hostname
- `fqdn` - A valid Fully Qualified Domain Name
- `deviceid` - A device ID
- `id` - The ID of the config to deploy to the specified device.
- `configtext` - The configuration text to deploy to the specified device.
- `start` - YYYY:MM:DD:HH:mm. The first date on which the task will run. Do not use this option with -sync.
- `sync` - Indicates that the command should return only after the deploy task is complete. Do not use this option with -start.
- `option` - current or startup_reload, as applicable to the device.

Examples

- `deploy config -ip 192.0.2.10 -id 1962 -sync -option current`
- `deploy config -ip "East Office:192.0.2.10" -id 1962 -sync -option current`
- `deploy config -ip 192.0.2.10 -id 276 -start 2004:02:29:23:59 -option startup_reload`
- `deploy config -ip 192.0.2.10 -configtext "logging 192.0.2.10\nlogging192.0.2.10" -option current`
**deploy image**

Deploy software images to device.

**Synopsis**

deploy image -ip <device ip address> -imageset <imageset name> -images <images separated by ,> [-reboot <reboot instruction>] [-rebootwait <reboot wait (in seconds)>] [-filesystem <file system of device>] [-pretask <task to run before deployment>] [-posttask <task to run after deployment>] [-verify <true|false>]

**Description**

Deploy software images to device.

- -ip - ip address of the device the images will deploy to.
- -imageset - imageset name the images from.
- -images - images from the imageset to be deployed.
- -reboot - wheather to reboot the device after deploy images.
- -rebootwait - seconds to wait before reboot.
- -filesystem - filesystem name of the device the images will deploy to.
- -pretask - name of task before deployment.
- -posttask - name of task after deployment.
- -verify - verify the image after deployment.

**Examples**

- deploy image -ip 192.0.2.10 -imageset fooset -images bar.bin,foo.bin -filesystem flash:
- deploy image -ip 192.0.2.10 -imageset fooset -images bar.bin,foo.bin -filesystem flash: -reboot -rebootwait 60
- deploy image -ip 192.0.2.10 -imageset fooset -images bar.bin,foo.bin -filesystem flash: -reboot -rebootwait 60 -posttask squeeze
- deploy image -ip 192.0.2.10 -imageset fooset -images bar.bin,foo.bin -filesystem flash: -verify true

**diff config**

Show the differences between two configs.

**Synopsis**

diff config -id1 <Config ID> -id2 <Config ID>

**Description**

- -id1 - The ID of a config
- -id2 - The ID of a config

**Examples**

- diff config -id1 1961 -id2 1989
disable device
Mark a device as disabled.

Synopsis
disable device [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>]

Description
- **-ip** - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- **-host** - A valid hostname
- **-fqdn** - A valid Fully Qualified Domain Name
- **-deviceid** - A device ID

Examples
- disable device -host rtr5.vfm.lab

discover driver
Discover a driver for a device.

Synopsis
discover driver [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>]

Description
Attempts to match a driver to the specified device.
- **-ip** - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.: The device for which a driver should be discovered.
- **-host** - A valid hostname: The device for which a driver should be discovered.
- **-fqdn** - A valid Fully Qualified Domain Name: The device for which a driver should be discovered.
- **-deviceid** - A device ID

Examples
- discover driver -ip 192.0.2.10
- discover driver -ip "East Site:192.0.2.10"
**discover drivers**
Discover drivers for all devices.

**Synopsis**
discover drivers

**Description**
Attempts to match a driver to each device that the system recognizes.

**Examples**
- discover drivers
- discover drivers -noskip

---

**enable device**
Mark a device as enabled.

**Synopsis**
enable device [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>]

**Description**
- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -host - A valid hostname
- -fqdn - A valid Fully Qualified Domain Name
- -deviceid - A device ID

**Examples**
- enable device -ip 192.0.2.10
- enable device -ip "East Site:192.0.2.10"

---

**exit**
Exit the system.

**Synopsis**
exit

**Description**
Enter the the system.

**Examples**
- exit
get snapshot

Get the config from a device.

Synopsis


Description

Get the config from a specified device either right away, or at some point in the future. The retrieval operation is actually a scheduled task. Using this command, you can set the task to repeat periodically.

- **-ip** - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- **-group** - A valid group name. Do not use this option with -ip (exactly one of -ip or -group must be specified).
- **-host** - A valid hostname
- **-fqdn** - A valid Fully Qualified Domain Name
- **-deviceid** - A device ID
- **-rep** - (#min | #:# | #days | #weeks | #months) where # is a positive integer. #:# is hours:minutes--the two integers do not have to be the same. Do not use this option with -sync.
- **-sync** - Indicates the command should return only after the snapshot retrieval task is complete. Do not use this option with -rep or -start.
- **-start** - YYYY:MM:DD:HH:mm. The first date on which the task will run.
- **-comment** - An optional comment about the snapshot.
- **-duration** - A number concatenated with a units signifier. Valid signifiers are m (minutes), h (hours), d (days), w (weeks). If this option is not provided, the duration for the task is set to 60 minutes.

Examples

- get snapshot -ip 192.0.2.10
- get snapshot -ip "East Office:192.0.2.10"
- get snapshot -ip 192.0.2.10 -sync
- get snapshot -group mygroup
import
Import device or device password information.

Synopsis
import -input <Filename or CSV data> -data <device or auth> [-log <Filename>] [-append true or false] [-discoverafter true or false] [-configuresyslog true or false] [-filter <Filename>] [-cleanafter true or false] [-deviceorigin <Any String>]

Description
This command can import into the system device or device password information contained in appropriately formatted CSV files. (Contact customer support for a CSV file format specification.)

- -input - Either the name of a file that contains CSV data or the CSV data itself.
- -data - Whether the type of information imported is devices or device authentication.
- -log - Command log file.
- -append - If true, will append imported information to existing information. If false, will overwrite existing device/auth records. This option is false by default.
- -discoverafter - Discover drivers for imported device? This option is false by default.
- -configuresyslog - Configure devices to send syslog messages to the system? Valid values are true | false
- -filter - An application that reads the input file from stdin, and writes a the system compatible CSV file to stdout.
- -cleanafter - If true, then after importing data, a process will run on the server that will delete old devices. Devices are deleted according to the current configuration of the system's "deletion-on-import" rules, and the argument to the deviceorigin option. This option is false by default.
- -deviceorigin - A description of the source of the data. This is recorded by the system, but is not visible via any UI.

Examples
• import -input devices.csv -data device -log import.log -append true -cleanafter false -deviceorigin "Border Routers" -filter prepro.exe
• import -input auth.csv -data auth -log import.log
• import -data device -input "primaryIPAddress\n1.2.3.4\n1.10.3.4"

list access
List all access records for a device.

Synopsis
list access [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-start <Date>] [-end <Date>]
Description

- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -host - A valid hostname
- -fqdn - A valid Fully Qualified Domain Name
- -deviceid - A device ID
- -start - Display only those access records created on or after the given date. Values for this option may be in one of the following formats: YYYY-MM-DD HH:MM:SS e.g. 2002-09-06 12:30:00 YYYY-MM-DD HH:MM e.g. 2002-09-06 12:30 YYYY-MM-DD e.g. 2002-09-06 YYYY/MM/DD e.g. 2002/09/06 YYYY:MM:DD:HH:MM e.g. 2002:09:06:12:30 Or, one of: now, today, yesterday, tomorrow, in the format: " " e.g. "3 days ago" is a positive integer. is one of: seconds, minutes, hours, days, weeks, months, years; is one of: ago, before, later, after.
- -end - Display only those access records created on or before the given date. Values for this option have the same format as for the option -start.

Examples

- list access -ip 192.0.2.10
- list access -ip "East Office:192.0.2.10"

list access all
List all access records for all devices.

Synopsis
list access all

Description

Examples

- list access all

list acl
List ACLs.

Synopsis
list acl [-host <Host Name>] [-ip <IP Address>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-aclid <ACL ID>] [-handle <Handle>] [-recent <Most Recent (true|false)>] [-includescript] [-includeapplication]

Description
Lists all device ACLs in the system unless you include any of the options to limit the ACLs listed.
-host - A valid host name.
-ip - List only ACLs with a valid IP Address (of format a.b.c.d where 0 <= a,b,c,d <= 255.)
-fqdn - List only ACLs with a valid Fully Qualified Domain Name
-deviceid - List only ACLs with this deviceid.
-aclid - List only ACLs with this aclid.
-handle - List only ACLs with this handle
-recent - Display only those acl's that are most recent.
-includescript - Include Script in the display.
-includeapplication - Include Application in the display.

Examples

- list acl
- list acl -ip 192.0.2.10
- list acl -ip 192.0.2.10 -aclid 139
- list acl -ip 192.0.2.10 -deviceid 201
- list acl -deviceid 501 -includescript
- list acl -handle test34 -recent true -includeapplication

list all drivers
List all drivers installed in the system.

Synopsis
list all drivers

Description

Examples

- list all drivers

list authentication
list Authentication.

Synopsis
list authentication [-rulename <Rule Name>]

Description
Displays the Authentication Rules by Rule Name.

- -rulename - List Authentication by rule name

Examples

- list authentication -rulename 1
list basicip
List all configs for which the BasicIP model can be shown.

Synopsis
list basicip [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-start <Date>] [-end <Date>]

Description

- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -host - A valid hostname
- -fqdn - A valid Fully Qualified Domain Name
- -deviceid - A device ID
- -start - Display only those configs stored on or after the given date. Values for this option may be in one of the following formats: YYYY-MM-DD HH:MM:SS e.g. 2002-09-06 12:30:00YYYY-MM-DD HH:MM e.g. 2002-09-06 12:30YYYY-MM-DD e.g. 2002-09-06YYYY/MM/DD e.g. 2002/09/06YYYY:MM:DD:HH:MM e.g. 2002:09:06:12:30Or, one of: now, today, yesterday, tomorrowOr, in the format: e.g. 3 days ago is a positive integer. is one of: seconds, minutes, hours, days, weeks, months, years; is one of: ago, before, later, after.
- -end - Display only those configs stored on or before the given date. Values for this option have the same format as for the option -start.

Examples

- list basicip -ip 192.0.2.10

list config
List all configs for the specified device.

Synopsis
list config [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-start <Date>] [-end <Date>] [-size] [-ids <Config ID List>]

Description

- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -host - A valid hostname
- -fqdn - A valid Fully Qualified Domain Name
- -deviceid - A device ID
• -start - Display only those configs stored on or after the given date. Values for this option may be in one of the following formats: YYYY-MM-DD HH:MM:SS e.g. 2002-09-06 12:30:00 YYYY-MM-DD HH:MM e.g. 2002-09-06 YYYY/MM/DD e.g. 2002/09/06 YYYY:MM:DD:HH:MM e.g. 2002:09:06:12:30 Or, one of: now, today, yesterday, tomorrow Or, in the format: e.g. 3 days ago is a positive integer. is one of: seconds, minutes, hours, days, weeks, months, years; is one of: ago, before, later, after.
• -end - Display only those configs stored on or before the given date. Values for this option have the same format as for the option -start.
• -size - Display the size (in bytes) of each config
• -ids - List only configs in this comma-separated list of IDs.

Examples

• list config -ip 192.0.2.10
• list config -ip "East Office:192.0.2.10"
• list config -ip 192.0.2.10 -size

list config all
List all configs for all devices.

Synopsis
list config all

Description

Examples

• list config all

list config id
List config IDs for the specified configs.

Synopsis
list config id [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-start <Date>] [-end <Date>] [-id <Config ID>]

Description

• -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
• -host - A valid hostname
• -fqdn - A valid Fully Qualified Domain Name
• -deviceid - A device ID
-start - Display only those configs stored on or after the given date. Values for this option may be in one of the following formats: YYYY-MM-DD HH:MM:SS e.g. 2002-09-06 12:30:00 YYYY-MM-DD HH:MM e.g. 2002-09-06 12:30 YYYY-MM-DD e.g. 2002-09-06 YYYY/MM/DD e.g. 2002/09/06 YYYY:MM:DD:HH:MM e.g. 2002:09:06:12:30 Or, one of: now, today, yesterday, tomorrow Or, in the format: e.g. 3 days ago is a positive integer. is one of: seconds, minutes, hours, days, weeks, months, years; . is one of: ago, before, later, after.

-end - Display only those configs stored on or before the given date. Values for this option have the same format as for the option -start.

-id - Display only the specified config id.

Examples

- list config id -ip 192.0.2.10
- list config id -ip "East Site:192.0.2.10"

list custom data definition

List Custom Data Definition.

Synopsis

list custom data definition -tablename <Table Name>

Description

List Custom Data Definition by table name.

- -tablename - List Custom Data for specific table

Examples

- list custom data definition -tablename "Device Configuration & Diagnostics"
- list custom data definition -tablename Devices
- list custom data definition -tablename "Device Blades/Modules"
- list custom data definition -tablename "Device Interfaces"
- list custom data definition -tablename "Device Groups"
- list custom data definition -tablename Users
- list custom data definition -tablename "User Groups"
- list custom data definition -tablename Tasks
- list custom data definition -tablename "Telnet/SSH Sessions"
**list device**

List devices.

**Synopsis**


**Description**

Lists all devices in the system unless you include any of the options to limit the devices listed.

- `-software` - List only devices running this software
- `-vendor` - List only devices with this vendor name
- `-type` - List only devices of this type (Router, Switch, etc.)
- `-model` - List only devices of this model ("2500 (3000 series)", BIG-IP, etc.)
- `-family` - List only devices in this device family ("Cisco IOS", F5, etc.)
- `-group` - List only devices in this device group
- `-disabled` - List only devices that are unmanaged.
- `-pollexcluded` - List only devices excluded from polling.
- `-ids` - List only devices in this comma-separated list of IDs.
- `-hierarchy` - List only devices in this hierarchy layer.
- `-host` - List only devices with this host name
- `-ip` - List only devices with this IP Address
- `-realm` - List only devices in this realm
- `-startid` - List devices starting with DeviceIDs greater than or equal to this one.
- `-limitcount` - Return this many rows (maximum defaults to 10000).

**Examples**

- list device
- list device -group "border routers"
- list device -group "border routers" -disabled
- list device -family "Cisco IOS"
- list device -vendor Nortel
- list device -ids 1023,763,8723

---

**list device data**

List configuration and diagnostic data records for the specified device.

**Synopsis**

list device data -ip <IP address> [-dataType <Data type>]

---
Description

- **-ip** - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- **-dataType** - A string describing the type of device data record to list

Examples

- list device data -ip 192.0.2.10
- list device data -ip 192.0.2.10 -dataType "configuration"

---

**list device family**

List device families.

**Synopsis**

list device family [-software <Software Version>] [-vendor <Device Vendor>] [-type <Device Type>] [-model <Device Model>] [-group <Device Group>]

**Description**

Lists device families in the system.

- **-software** - List only device families for devices running this software
- **-vendor** - List only device families for devices with this vendor name
- **-type** - List only device families for devices of this type (Router, Switch, etc.)
- **-model** - List only device families for devices of this model ("2500 (3000 series)", BIG-IP, etc.)
- **-group** - List only device families for devices in this device group

**Examples**

- list device family
- list device family -group "border routers"
- list device family -vendor Nortel

---

**list device group**

List device groups.

**Synopsis**


**Description**

Lists device groups in the system.

- **-software** - List only device groups for devices running this software
• -vendor - List only device groups for devices with this vendor name
• -type - List only device groups for devices of this type (Router, Switch, etc.)
• -model - List only device groups for devices of this model ("2500 (3000 series)", BIG-IP, etc.)
• -family - List only device groups for devices in this device family ("Cisco IOS", F5, etc.)
• -parent - List only device groups that are direct descendants of this parent device group name

Examples

• list device group
• list device group -family "Cisco IOS"
• list device group -vendor Nortel

list device id
list device IDs.

Synopsis

Description
Lists all device IDs in the system unless you include any of the options to limit the device IDs listed.

• -software - List only devices running this software
• -vendor - List only devices with this vendor name
• -type - List only devices of this type (Router, Switch, etc.)
• -model - List only devices of this model ("2500 (3000 series)", BIG-IP, etc.)
• -family - List only devices in this device family ("Cisco IOS", F5, etc.)
• -group - List only devices in this device group
• -disabled - List only devices that are unmanaged.
• -pollexcluded - List only devices excluded from polling.
• -id - List only this device.
• -host - List only devices with this host name
• -ip - List only devices with this IP Address
• -realm - List only devices in this realm
• -hierarchy - List only devices with this hierarchy layer
• -viewable_by - List only devices that are viewable by this user

Examples

• list device id
• list device id -group "border routers"
list device model
List device model names.

Synopsis
list device model [-software <Software Version>] [-vendor <Device Vendor>] [-type <Device Type>] [-family <Device Family>] [-group <Device Group>]

Description
Lists device model names in the system.

-examples
- software - List only device model names for devices running this software
- vendor - List only device model names for devices with this vendor name
- type - List only device model names for devices of this type (Router, Switch, etc.)
- family - List only device model names for devices in this device family ("Cisco IOS", F5, etc.)
- group - List only device model names for devices in this device group

Examples

- list device model
- list device model -group "border routers"
- list device model -family "Cisco IOS"
- list device model -vendor Nortel

list device software
List device software versions.

Synopsis
list device software [-vendor <Device Vendor>] [-type <Device Type>] [-model <Device Model>] [-family <Device Family>] [-group <Device Group>]

Description
Lists device software versions in the system.

-examples
- vendor - List only device software versions for devices with this vendor name
- type - List only device software versions for devices of this type (Router, Switch, etc.)
- model - List only device software versions for devices of this model ("2500 (3000 series)", BIG-IP, etc.)
-family - List only device software versions for devices in this device family
("Cisco IOS", F5, etc.)
-group - List only device software versions for devices in this device group

Examples

- list device software
- list device software -group "border routers"
- list device software -family "Cisco IOS"
- list device software -vendor Nortel

list device type
List device types.

Synopsis
list device type [-software <Software Version>] [-vendor <Device Vendor>] [-model
<Device Model>] [-family <Device Family>] [-group <Device Group>]

Description
Lists device types in the system.

- -software - List only device types for devices running this software
- -vendor - List only device types for devices with this vendor name
- -model - List only device types for devices of this model ("2500 (3000 series)",
BIG-IP, etc.)
- -family - List only device types for devices in this device family ("Cisco IOS", F5,
etc.)
- -group - List only device types for devices in this device group

Examples

- list device type
- list device type -group "border routers"
- list device type -family "Cisco IOS"
- list device type -vendor Nortel

list device vendor
List device manufacturers.

Synopsis
list device vendor [-software <Software Version>] [-type <Device Type>] [-model
<Device Model>] [-family <Device Family>] [-group <Device Group>]
Description
Lists device manufacturers in the system.

- software - List only device manufacturers for devices running this software
- type - List only device manufacturers for devices of this type (Router, Switch, etc.)
- model - List only device manufacturers for devices of this model ("2500 (3000 series)", BIG-IP, etc.)
- family - List only device manufacturers for devices in this device family ("Cisco IOS", F5, etc.)
- group - List only device manufacturers for devices in this device group

Examples
- list device vendor
- list device vendor -group "border routers"
- list device vendor -family "Cisco IOS"

list deviceinfo
List all configs for which the DeviceInformation model can be shown.

Synopsis
list deviceinfo [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>]

Description
- ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- host - A valid hostname
- fqdn - A valid Fully Qualified Domain Name
- deviceid - A device ID

Examples
- list deviceinfo -ip 192.0.2.10

list diagnostic
List all configs for which the given diagnostic may be shown.

Synopsis
list diagnostic -diagnostic <Diagnostic Name> [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-start <Date>] [-end <Date>]
Description

- **-diagnostic** - A diagnostic name
- **-ip** - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- **-host** - A valid hostname
- **-fqdn** - A valid Fully Qualified Domain Name
- **-deviceid** - A device ID
- **-start** - Display only those diagnostics stored on or after the given date. Values for this option may be in one of the following formats: YYYY-MM-DD HH:MM:SS e.g. 2002-09-06 12:30:00 YYYY-MM-DD HH:MM e.g. 2002-09-06 12:30 YYYY-MM-DD e.g. 2002-09-06 YYYY/MM/DD e.g. 2002/09/06 YYYY:MM:DD:HH:MM e.g. 2002:09:06:12:30 Or, one of: now, today, yesterday, tomorrow Or, in the format: e.g. 3 days ago is a positive integer. is one of: seconds, minutes, hours, days, weeks, months, years.; is one of: ago, before, later, after.
- **-end** - Display only those diagnostics created on or before the given date. Values for this option have the same format as for the option -start.

Examples

- list diagnostic -ip 192.0.2.10 -diagnostic "vlan report"

---

**list drivers**

List all drivers associated with a device.

Synopsis

list drivers [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>]

Description

- **-ip** - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- **-host** - A valid hostname
- **-fqdn** - A valid Fully Qualified Domain Name
- **-deviceid** - A device ID

Examples

- list drivers -ip 192.0.2.10

---

**list event**

List events.
Synopsis

list event [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-type <Event Type>] [-start <Date>] [-end <Date>]

Description

Lists all the events and system messages

- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in. Display only those events associated with the specified device.
- -host - A valid hostname: Display only those events associated with the specified device.
- -fqdn - A valid Fully Qualified Domain Name: Display only those events associated with the specified device.
- -deviceid - A device ID
- -type - A valid event type: Display only events of this type. Values for this option may one of the following:
  Approval No Longer Required
  Approval Request
  Approval Granted
  Approval Task Changed
  Approval Task Deleted
  Approval Denied
  Approval Task Timeout
  Approval Override
  Authorization Error
  User Authentication Error
  Configuration Policy Added
  Configuration Policy Pattern Timeout
  Configuration Policy Non-Compliance
  Configuration Policy Changed
  Configuration Policy Pattern Timeout
  Configuration Rule Added
  Configuration Rule Changed
  Device Access Failure
  Device Added
  Device Booted
  Device Command Script Failed
  Device Command Script Completed Successfully
  Device Configuration Change - No User
  Device Configuration Deployment Failure
  Device Configuration Deployment Device Data Failure
  Device Deleted
  Device Diagnostic Changed
  Device Diagnostic Failed
  Device Diagnostic Completed
  Device Flash Storage Running Low
  Group Added
  Group Deleted
  Device Inaccessible
  Last Used Device Password Changed
  Device Managed
  Device Missing from Import
  Device Permissions - Modified
  Device Reservation Conflict
  Device Snapshot
  Software Change
  Device Startup/Running Config Difference
  Device Unmanaged
  Software Vulnerability Detected
  Email Report Saved
  External Directory Server Authentication Error
  License Almost Exceeded
  License Almost Expired
  License Exceeded
  License Expired
  Module Added
  Module Changed
  Module Removed
  Monitor Okay
  Monitor Error
  Device Permissions - New
  Device Password Change Failure
  Concurrent Telnet/SSH Session
  Override Reserved Device Configuration
  Change Scheduled for Deploy
  Configuration Edited Scheduled for Deploy
  Password Modified
  Startup Session Data Captured
  Software Update Failed
  Software Update Succeeded
  Summary Reports Generated
  Pending Task Deleted
  Task Started
  Ticket Created
  User Login
  User Logout
  User Added
  User Deleted
  User Message
-start - Display only events after this date. Values for this option may be in one of the following formats: YYYY-MM-DD HH:MM:SS e.g. 2002-09-06 12:30:00 YYYY-MM-DD HH:MM e.g. 2002-09-06 12:30 YYYY-MM-DD e.g. 2002-09-06 YYYY/MM/DD e.g. 2002/09/06 YYYY:MM:DD:HH:MM e.g. 2002:09:06:12:30 Or, one of: now, today, yesterday, tomorrow Or, in the format: e.g. 3 days ago is a positive integer. is one of: seconds, minutes, hours, days, weeks, months, years.; is one of: ago, before, later, after.

-end - Display only events before this date.

Examples

- list event -ip 192.0.2.10
- list event -ip "East Site:192.0.2.10"
- list event -start yesterday

list group id
List all device groups or user groups viewable by userID.

Synopsis
list group id -type <Type> [-viewable_by <Viewable By>]

Description
List all device groups or user groups viewable by a particular user.

- -type - Type
- -viewable_by - Viewable By

Examples

- list group id -type device -viewable_by 201
- list group id -type user -viewable_by 201

list groups
List groups of the specified type; for a specific device or all groups in the system.

Synopsis
list groups -type <Type> [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-parent <Parent Group Name>]

Description

- -type - The type of the groups to be listed. "device" is currently the only valid argument to this option.
- -ip - List all device groups containing the device with this IP address
- -host - List all device groups containing the device with this hostname

PERL API Reference Guide 69
• -fqdn - List all device groups containing the device with this Fully Qualified Domain Name
• -deviceid - List all device groups containing the device with this device ID
• -parent - List all device groups that are children of the indicated parent group

Examples

• list groups -type device
• list groups -type device -ip 192.0.2.10
• list groups -type device -parent "North America"

(list icmp
List all configs for which the ICMPTest model may be shown.

Synopsis
list icmp [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-start <Date>] [-end <Date>]

Description

• -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
• -host - A valid hostname
• -fqdn - A valid Fully Qualified Domain Name
• -deviceid - A device ID
• -start - Display only those ICMPTest models stored on or after the given date. Values for this option may be in one of the following formats: YYYY-MM-DD HH:MM:SS e.g. 2002-09-06 12:30:00 YYYY-MM-DD HH:MM e.g. 2002-09-06 12:30 YYYY-MM-DD e.g. 2002-09-06 YYYY/MM/DD e.g. 2002/09/06 YYYY:MM:DD:HH:MM e.g. 2002:09:06:12:30 Or, one of: now, today, yesterday, tomorrow Or, in the format: e.g. 3 days ago is a positive integer. is one of: seconds, minutes, hours, days, weeks, months, years; is one of: ago, before, later, after.
• -end - Display only those ICMPTest models stored on or before the given date. Values for this option have the same format as for the option -start.

Examples

• list icmp -ip 192.0.2.10

(list image
List images in database or device.

Synopsis
list image [-ip <ip address>] [-imageset <imageset name>]

70  PERL API Reference Guide
Description
Use -imageset option to list images in database, and -ip to list images in device

- ip - The device ip which the images list from.
- -imageset - The imageset which images list from.

Examples

- list image -ip 10.1.1.1
- list image -imageset fooset

list imageoption
List information for device which is not under management but in configuration data.

Synopsis
list imageoption -name <device property name (model|processor|bootrom)>

Description
List information for device which is not under management but in configuration data for software image management purpose.

- -name - device property name to list.

Examples

- list imageoption -name model

list imageset
List imageset in database.

Synopsis
list imageset

Description
List imageset in database

Examples

- list imageset

list int
List all configs for which the ShowInterfaces model may be shown.
Synopsis
list int [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-start <Date>] [-end <Date>]

Description

- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -host - A valid hostname
- -fqdn - A valid Fully Qualified Domain Name
- -deviceid - A device ID
- -start - Display only those ShowInterfaces models stored on or after the given date. Values for this option may be in one of the following formats: YYYY-MM-DD HH:MM:SS e.g. 2002-09-06 12:30:00 YYYY-MM-DD HH:MM e.g. 2002-09-06 12:30 YYYY-MM-DD e.g. 2002-09-06 YYYY/MM/DD e.g. 2002/09/06 YYYY:MM:DD:HH:MM e.g. 2002:09:06:12:30 Or, one of: now, today, yesterday, tomorrow Or, in the format: e.g. 3 days ago is a positive integer. is one of: seconds, minutes, hours, days, weeks, months, years; is one of: ago, before, later, after.
- -end - Display only those ShowInterfaces models stored on or before the given date. Values for this option have the same format as for the option -start.

Examples

- list int -ip 192.0.2.10

list ip
List ip.

Synopsis
list ip [-deviceip <Device IP address>] [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>]

Description
Lists ip addresses for specific device.

- -deviceip - The device's ip address a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -host - A valid hostname
- -fqdn - A valid Fully Qualified Domain Name
- -deviceid - A device ID

Examples

- list ip -deviceip 192.0.2.10
list ip all
List all secondary ip.

Synopsis
list ip all

Description
List all secondary ip addresses in the system.

Examples
• list ip all

list metadata
List all the custom data associated with a specific entry in a specific table.

Synopsis
list metadata -table <Database Table> -associatedtableid <Matching Row ID>

Description
• -table - Table the data is associated with
  • -associatedtableid - ID of the associated row from the table.

Examples
• list metadata -table RN_DEVICE -associatedtableid 21031
• list metadata -table RN_DEVICE_PORT -associatedtableid 221

list metadata field
List all the custom data fields associated with a specific table.

Synopsis
list metadata field -table <Database Table>

Description
• -table - Table the fields are associated with

Examples
• list metadata field -table RN_DEVICE
• list metadata field -table RN_DEVICE_PORT
list module
List modules (or blades) in the system.

Synopsis

Description
- -model - List only device modules matching this model number
- -type - List only device modules matching this module description
- -firmware - List only device modules matching this firmware version
- -hardware - List only device modules matching this hardware revision
- -memory - List only device modules with this amount of memory
- -comment - List only device modules matching this comment
- -ip - List only device modules on the device with this IP address
- -host - List only device modules on the device with this hostname
- -fqdn - List only device modules on the device with this Fully Qualified Domain Name
- -deviceid - List only device modules on the device with this device ID
- -group - List only device modules on all devices with this device group name

Examples
- list module -host border7.red
- list module -type ethernet

list ospfneighbor
List all configs for which the ShowOSPFNeighbors model may be shown.

Synopsis
list ospfneighbor [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-start <Date>] [-end <Date>]

Description
- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -host - A valid hostname
- -fqdn - A valid Fully Qualified Domain Name
- -deviceid - A device ID
• -start - Display only those ShowOSPFNeighbors models stored on or after the
  given date. Values for this option may be in one of the following formats:
  YYYY-MM-DD HH:MM:SS e.g. 2002-09-06 12:30:00
  YYYY-MM-DD HH:MM e.g. 2002-09-06 12:30
  YYYY-MM-DD e.g. 2002-09-06
  YYYY/MM/DD e.g. 2002/09/06
  YYYY:MM:DD:HH:MM e.g. 2002:09:06:12:30
  Or, one of: now, today, yesterday, tomorrow
  Or, in the format: e.g. 3 days ago is a positive integer. is one of:
  seconds, minutes, hours, days, weeks, months, years;
  is one of: ago, before, later, after.
• -end - Display only those ShowOSPFNeighbors models stored on or before the
  given date. Values for this option have the same format as for the option -start.

Examples

  • list ospfneighbor -ip 192.0.2.10

list partition
Show details for a single view.

Synopsis
list partition -viewname <View Name>

Description
Show details for a single partition.

  • -viewname - The View Name to show.

Examples

  • list partition -viewname Site

list policy id
Lists IDs of all policies that apply to a given device

Synopsis
list policy id -deviceid <Device ID>

Description

  • -deviceid - device id

Examples

  • list policy id -deviceid 312
list policy rule

Lists all rules of a policy

**Synopsis**

list policy rule -policyid <Policy ID>

**Description**

- -policyid - policy id

**Examples**

- list policy rule -policyid 6120

---

list port

List ports (or interfaces) for a specific device in the system.

**Synopsis**

list port [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>]

**Description**

- -ip - List all device ports on the device with this IP address
- -host - List all device ports on the device with this hostname
- -fqdn - List all device ports on the device with this Fully Qualified Domain Name
- -deviceid - List all device ports on the device with this device ID

**Examples**

- list port -host border7.red
- list port -ip 192.0.2.10

---

list routing

List all routing tables for a device.

**Synopsis**

list routing [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-start <Date>] [-end <Date>]

**Description**

- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -host - A valid hostname
- -fqdn - A valid Fully Qualified Domain Name
- -deviceid - A device ID
-start - Display only those routing tables stored on or after the given date. Values for this option may be in one of the following formats: YYYY-MM-DD HH:MM:SS e.g. 2002-09-06 12:30:00 YYYY-MM-DD HH:MM e.g. 2002-09-06 12:30 YYYY-MM-DD e.g. 2002-09-06 YYYY/MM/DD e.g. 2002/09/06 YYYY:MM:DD:HH:MM e.g. 2002:09:06:12:30 Or, one of: now, today, yesterday, tomorrow Or, in the format: e.g. 3 days ago is a positive integer. is one of: seconds, minutes, hours, days, weeks, months, years; is one of: ago, before, later, after.
-end - Display only those routing tables stored on or before the given date. Values for this option have the same format as for the option -start.

Examples

- list routing -ip 192.0.2.10
- list routing -ip "East Site:192.0.2.10"

list rule condition
Lists all conditions of a policy rule

Synopsis
list rule condition -ruleid <Rule ID>

Description

- -ruleid - rule id

Examples

- list rule condition -ruleid 6120

list script
List command scripts, advanced scripts and/or diagnostics.

Synopsis
list script [-type <Type>] [-scripttype <Script Type>] [-name <Name>] [-mode <Mode>] [-ids <Script ID List>]

Description

- -type - Type of the desired script or diagnostic - may be command, advanced or diagnostic
- -scripttype - User defined script type (i.e. subcategory) - applies only to command scripts and advanced scripts
- -name - Script name
- -mode - Script mode - for command scripts and diagnostics the script's level of device access (such as Cisco IOS enable); for advanced scripts the device family (such as Cisco IOS)
- -ids - List only scripts in this comma-separated list of IDs.
Examples

- list script
- list script -type diagnostic
- list script -type advanced -scripttype "Core Provisioning Scripts"
- list script -name "Set Banner"
- list script -mode "Cisco IOS enable"

list script id
List command script IDs, advanced scripts and/or diagnostics.

Synopsis
list script id [-type <Type>] [-scripttype <Script Type>] [-name <Name>] [-mode <Mode>] [-id <ID>]

Description
- -type - Type of the desired script or diagnostic - may be command, advanced or diagnostic
- -scripttype - User defined script type (i.e. subcategory) - applies only to command scripts and advanced scripts
- -name - Script name
- -mode - Script mode - for command scripts and diagnostics the script's level of device access (such as Cisco IOS enable); for advanced scripts the device family (such as Cisco IOS)
- -id - Script ID

Examples

- list script id
- list script id -type diagnostic
- list script id -type advanced -scripttype "Core Provisioning Scripts"
- list script id -name "Set Banner"
- list script id -mode "Cisco IOS enable"

list script mode
List valid modes for commands scripts and diagnostics, and valid device families for advanced scripts, for all devices or a specified device.

Synopsis
list script mode [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-id <Device ID>] [-type <Type>]

Description
- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
list session
List all interceptor log records for a device.

Synopsis
list session [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-start <Date>] [-end <Date>]

Description

- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -host - A valid hostname
- -fqdn - A valid Fully Qualified Domain Name
- -deviceid - A device ID
- -start - Display only those interceptor log records created on or after the given date. Values for this option may be in one of the following formats: YYYY-MM-DD HH:MM:SS e.g. 2002-09-06 12:30:00 YYYY-MM-DD HH:MM e.g. 2002-09-06 12:30 YYYY-MM-DD e.g. 2002-09-06 YYYY/MM/DD e.g. 2002/09/06 YYYY:MM:DD:HH:MM e.g. 2002:09:06:12:30 Or, one of: now, today, yesterday, tomorrowOr, in the format: e.g. 3 days ago is a positive integer. is one of: seconds, minutes, hours, days, weeks, months, years;. is one of: ago, before, later, after.
- -end - Display only those interceptor log records created on or before the given date. Values for this option have the same format as for the option -start.

Examples

- list session -ip 192.0.2.10
list site
List all Sites in the system.

Synopsis
list site

Description
Result includes the name of each site in the system, and the number of devices in each site.

Examples
- list site

list sys oids all
List all sys oids supported by the system.

Synopsis
list sys oids all

Description
List all sys oids in the system.

Examples
- list sys oids all

list system message
List system messages.

Synopsis
list system message [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-start <Date>] [-end <Date>]

Description
Lists all system messages unless you include one of the options. Including one of the device options displays all system messages associated with the specified device.

- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -host - A valid hostname
- -fqdn - A valid Fully Qualified Domain Name
- -deviceid - A device ID
-start - Display only those system messages created on or after the given date. Values for this option may be in one of the following formats: YYYY-MM-DD HH:MM:SS e.g. 2002-09-06 12:30:00 YYYY-MM-DD HH:MM e.g. 2002-09-06 12:30 YYYY-MM-DD e.g. 2002-09-06 YYYY/MM/DD e.g. 2002/09/06 YYYY:MM:DD:HH:MM e.g. 2002:09:06:12:30 Or, one of: now, today, yesterday, tomorrow Or, in the format: e.g. 3 days ago is a positive integer. is one of: seconds, minutes, hours, days, weeks, months, years; is one of: ago, before, later, after.

-end - Display only those system messages created on or before the given date. Values for this option have the same format as for the option -start.

Examples

- list system message
- list system message -host chi-border-07

list task
Display a list of tasks.

Synopsis
list task [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-start <Task start date>] [-end <Task end date>] [-parentid <Parent task ID>] [-status <Task status>] [-id <Task ID>]

Description
This command behaves differently depending on the options you give it. The command by itself returns a list of all tasks. Each option filters the returned list of tasks, causing it to return a subset of the total list.

- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.: Display only those tasks associated with the specified device.
- -host - A valid hostname: Display only those tasks associated with the specified device.
- -fqdn - A valid Fully Qualified Domain Name: Display only those tasks associated with the specified device.
- -deviceid - A valid device ID: Display only those tasks associated with the specified device.
- -start - YYYY:MM:DD:HH:mm: Display only those tasks whose schedule date falls on or after the given date.
- -end - YYYY:MM:DD:HH:mm: Display only those tasks whose schedule date falls on or before the given date.
- -parentid - a task ID: Display only those tasks whose parent is the task specified by the given Task ID.
- -status - (pending | succeeded | failed | running | paused | starting | waiting | synchronous | skipped | warning): Display only those tasks with the specified status.
- -id - a task ID: Display the task with the given task ID.
Examples

- list task -parentid 78
- list task -start 2004:02:29:23:59 -status failed
- list task -start 2004:02:29:00:00 -end 2004:03:01:00:00 -ip 192.0.2.10
- list task -id 23

list task all
List all tasks.

Synopsis
list task all

Description
Equivalent to "list task".

Examples

- list task all

list topology
List Device Information from Topology Data.

Synopsis
list topology [-mac <MAC Address>] [-ip <IP Address>]

Description

- -mac - Show only devices that have seen this MAC address (no colons)
- -ip - Show only devices that have seen this IP Address

Examples

- list topology
- list topology -mac 00AABBCCDDEE
- list topology -ip 192.0.2.10
- list topology -mac 00AABBCCDDEE -ip 192.0.2.10

list topology graph
List probable Layer 1 topology data.

Synopsis
list topology graph [-deviceids <List of Device IDs>] [-deviceportids <List of Device Port IDs>] [-serverids <List of Server IDs>] [-serverportids <List of Server Interface IDs>] [-deviceid <A Device ID>]
Description

- **-deviceids** - A comma separated list of device IDs
- **-deviceportids** - A comma separated list of device port IDs
- **-serverids** - A comma separated list of server IDs
- **-serverportids** - A comma separated list of server interface IDs
- **-deviceid** - A device ID

Examples

- `list topology graph -deviceid 193`
- `list topology graph -deviceids 54302,16001`

---

**list topology ip**

List IP address Topology Data.

**Synopsis**

`list topology ip [-deviceip <Device IP/hostname>] [-portid <Device Port ID>] [-notcurrent < >] [-type <(all|internal|connected)>] [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>]`

**Description**

- **-deviceip** - The ip/hostname of the device for which to show IP topology data.
- **-portid** - The port id for which to show IP topology data.
- **-notcurrent** - Specify to limit output to only IP topology data that is no longer visible.
- **-type** - Limit the IP data to a specific type.
- **-ip** - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- **-host** - A valid hostname
- **-fqdn** - A valid Fully Qualified Domain Name
- **-deviceid** - A device ID

**Examples**

- `list topology ip -deviceip 192.0.2.10`
- `list topology ip -portid 54302`
- `list topology ip -deviceip 192.0.2.10 -notcurrent`
- `list topology ip -portid 54302 -type internal`
- `list topology ip -deviceid 1401`
list topology mac
List MAC address Topology Data.

Synopsis
list topology mac [-deviceip <Device IP/hostname>] [-portid <Device Port ID>] [-notcurrent < >] [-type <(all|internal|connected)>] [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>]

Description
- -deviceip - The ip/hostname of the device for which to show MAC topology data.
- -portid - The port id for which to show MAC topology data.
- -notcurrent - Specify to limit output to only MAC topology data that is no longer visible.
- -type - Limit the MAC data to a specific type.
- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -host - A valid hostname
- -fqdn - A valid Fully Qualified Domain Name
- -deviceid - A device ID

Examples
- list topology mac -deviceip 192.0.2.10
- list topology mac -portid 54302
- list topology mac -deviceid 192.0.2.10 -notcurrent
- list topology mac -deviceip 192.0.2.10 -notcurrent
- list topology mac -portid 54302 -type internal
- list topology mac -deviceid 1401

list user
List all users.

Synopsis
list user

Description
List all users known to the system.

Examples
- list user

list user id
List all users viewable by userID.

Synopsis
list user id [-viewable_by <Viewable By>]

84 PERL API Reference Guide
Description
List all users viewable by a particular user.

- -viewable_by - Viewable By

Examples
- list user id -viewable_by 201

list view
display the views defined within the system.

Synopsis
list view

Description
Show the views defined within the system. This command takes no arguments.

Examples
- list view

list vlan
Show the vlans and their associated Device Port ID on a device.

Synopsis
list vlan [-deviceip <Device IP/Hostname>] [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>]

Description
- -deviceip - The ip/hostname of the device for which to show vlans.
- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -host - A valid hostname
- -fqdn - A valid Fully Qualified Domain Name
- -deviceid - A device ID

Examples
- list vlan -deviceip 192.0.2.10
- list vlan -deviceid 1401
**list vlan ports**
Show the ports on a given vlan, identified by its port id.

**Synopsis**
list vlan ports -id <VLAN port id>

**Description**
- -id - The port id of the vlan (provided in 'list vlan').

**login**

**Synopsis**
login -username <Username> -password <Password> [-host <Host>]

**Description**
- -username - @ProductAbbreviation@ username
- -password - @ProductAbbreviation@ password
- -host - URL of @ProductAbbreviation@ server (defaults to localhost:1099)

**logout**

**Synopsis**
logout -sessionid <Session ID>

**Description**
- -sessionid - @ProductAbbreviation@ SOAP API Session ID

**mod advanced script**
Modify an existing advanced script.

**Synopsis**

**Description**
Modify the indicated advanced script. The desired script can be specified by ID or name. If more than one name match occurs, then an error will be reported and you must specify the unique script desired by ID.

- -id - ID of the advanced script to edit
- -name - Name of the advanced script to edit
-newname - New name for the script being modified
-description - New description for the script being modified
-scripttype - New script type (i.e. user defined subcategory)
-family - New device family for the script being modified
-language - New language for the script being modified - must be a supported language such as Expect or Perl
-parameters - New command line parameters for the script being modified
-script - New script text

Examples

- mod advanced script -id 22 -newname "Set Duplex" -description "Sets the interface duplex configuration" -scripttype "Interface Management Scripts"
- mod advanced script -name "Extended Ping" -family "Cisco IOS" -language "Expect" -parameters "-l /usr/etc/log.txt" -script "send \"extended ping $Target_IP$\""

mod authentication

Modify device password information.

Synopsis


Description

This command can modify passwords on a specific device, across all devices in a device group, or merely update what the system knows of the device's password information. When using this command to modify passwords on a device or device group, the modification operation is actually a scheduled task.

- **-loc** - The location to which password information should be written. Valid values for this argument are "db", "device", and "group". "db" tells the command that password information should be changed only in the system's database. "device" tells the command that the password changes should be made on the device as well and "group" performs the same function as "device" but across all devices in the group.
- **-ip** - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE where SITE is the name of the Site the device is in.: An existing device to which this password information should apply.
- **-host** - A valid hostname: An existing device to which this password information should apply.
-fqdn - A valid Fully Qualified Domain Name: An existing device to which this password information should apply.
-deviceid - A valid device ID: An existing device to which this password information should apply.
-snmpro - When used in conjunction with -loc db, this argument is taken as a single community string understood by the system as THE read only community string for the device or network. When used in conjunction with -loc device, this argument is taken as a comma-separated list of read only community strings to be, either set on the device, or appended to an existing list of read only community strings (depends on whether or not the -appendsnmpro flag was supplied.)
-snmprw - When used in conjunction with -loc db, this argument is taken as a single community string understood by the system as THE read write community string for the device or network. When used in conjunction with -loc device, this argument is taken as a comma-separated list of read write community strings to be, either set on the device, or appended to an existing list of read write community strings (depends on whether or not the -appendsnmprw flag was supplied.)
-snmpv3user - When used in conjunction with -loc db, this argument is taken as the username for snmpv3 access.
-snmpv3authpw - When used in conjunction with -loc db, this argument is taken as the authentication password for snmpv3 access.
-snmpv3encryptpw - When used in conjunction with -loc db, this argument is taken as the encryption password for snmpv3 access.
-user - Username.
-passwd - Password.
-enableuser - ADDITIONAL username to get to "enable" mode.
-enablepasswd - ADDITIONAL password to get to "enable" mode.
-connectionmethods - The methods used by the system to connect to devices. Can be telnet, serial_direct, or SSH.
-accessvariables - To override variables in the script, such as prompts.
-start - YYYY:MM:DD:HH:mm. The first date on which the task will run. Use this option only if the argument to the -loc flag is "device".
-appendsnmpro - Supply this option if read only community strings should be appended to any existing on the device. Use this option only if the argument to the -loc flag is "device".
-appendsnmprw - Supply this option if read write community strings should be appended to any existing on the device. Use this option only if the argument to the -loc flag is "device".
-sync - Indicates that the command should return only after the password change task is complete. Do not use this option with -start.
-group - The group name for performing this command across all devices in a group.
-rulename - The password rule name to apply the access variables to

Examples

```
mod authentication -loc db -ip 192.0.2.10 -passwd fish -snmpro public -enablepasswd 31337
mod authentication -loc device -ip 192.0.2.10 -passwd limited -enablepasswd full
```
mod command script

Modify an existing command script.

**Synopsis**


**Description**

Modify the indicated command script. The desired script can be specified by ID or name. If more than one name match occurs, then an error will be reported and you must specify the unique script desired by ID.

- **-id** - ID of the command script to edit
- **-name** - Name of the command script to edit
- **-newname** - New name for the script being modified
- **-description** - New description for the script being modified
- **-scripttype** - New script type (i.e. user defined subcategory)
- **-mode** - New command script mode
- **-driver** - New list of applicable drivers - provided as a comma separated list of internal driver names
- **-script** - New script text

**Examples**

- mod command script -id 22 -newname "Set Duplex" -description "Sets the interface duplex configuration" -scripttype "Interface Management Scripts"
- mod command script -name "Extended Ping" -mode "Cisco IOS enable" -driver "CiscoIOSGeneric,CiscoI OSSwitch" -script "extended ping $Target_IP$"
mod device
Modify the properties of a device.

Synopsis

Description

- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -host - A valid hostname
- -fqdn - A valid Fully Qualified Domain Name
- -deviceid - A device ID
- -hostname - The device's new host name
- -comment - Additional information regarding the device.
- -description - The descriptive name of the device (informational only).
- -model - The device's model (such as 2620).
- -vendor - The device's vendor (such as Cisco).
- -domain - A fully qualified domain name (such as www.google.com).
- -serial - The device's serial number.
- -asset - The device's asset tag.
- -location - The device's location.
- -unmanaged - 0: Mark this device as managed by the system. 1: Mark this device to be unmanaged by the system.
- -nopoll - 0: Mark this device to be polled for changes. 1: Mark this device as not to be polled for changes.
- -newIP - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device will be put in. This is the new IP address of the device.
- -consoleip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device will be put in. This is the new IP address of the device.
- -consoleport - The port number
- -tftpserverip - a.b.c.d where 0 <= a,b,c,d <= 255
- -natip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with REALM_NAME:, where REALM_NAME is the name of the Realm the address is in.
- -customname - The custom field name
- -customvalue - The custom field value
- -useconsoleserver - true, if the device uses a console server. false, if the device does not.
-accessmethods - A comma-separated list of access methods, or "none". The set of access methods: {telnet, ssh, rlogin, SCP, FTP, TFTP, SNMP, snmp_noauthnopriv, snmp_authnopriv, snmp_authpriv).

-hierarchylayer - This device attribute is used in diagramming. When you config a network diagram, you can select which hierarchy layers on which to filter. Valid values include: (core, distribution, access, edge and "layer not set").

Examples

- mod device -ip 192.0.2.10 -newIP 192.0.2.10
- mod device -ip 192.0.2.10 -newIP "West Site:192.0.2.10"
- mod device -ip "East Site:192.0.2.10" -newIP "West Site:192.0.2.10"
- mod device -ip 192.0.2.10 -nopoll 1 -comment "enabled polling by change detection."
- mod device -ip 192.0.2.10 -customname Owner -customvalue Bob
- mod device -ip 192.0.2.10 -useconsolesever false

mod diagnostic
Modify an existing custom diagnostic script.

Synopsis


Description

Modify the indicated diagnostic script. The desired diagnostic can be specified by ID or name. If more than one name match occurs, then an error will be reported and you must specify the unique diagnostic desired by ID.

- -id - ID of the diagnostic to edit
- -name - Name of the diagnostic to edit
- -newname - New name for the diagnostic being modified
- -description - New description for the diagnostic being modified
- -mode - New command script mode
- -driver - New list of applicable drivers - provided as a comma separated list of internal driver names
- -script - New diagnostic script text

Examples

- mod diagnostic -id 22 -newname "Show IP CEF" -description "Gather IP CEF information"
- mod diagnostic -name "Extended Ping To Core" -mode "Cisco IOS enable" -driver "CiscoIOSGeneric,CiscoIOSSwitch" -script "extended ping 192.0.2.10"
**mod group**
Modify a group.

**Synopsis**

```
mod group -type <Type> -name <Name> [-newname <New name>] [-comment <Comment>] [-customname <Customname>] [-customvalue <Customvalue>] [-shared <Shared>]
```

**Description**
Modify the comments associated with and/or the name of a group.

- **-type** - The type of the group. "device" is currently the only valid argument to this option.
- **-name** - The name of the group to be modified.
- **-newname** - The new name for the modified group. Do not use this option unless you also use -name.
- **-comment** - Additional information regarding the group.
- **-customname** - The custom field name
- **-customvalue** - The custom field value
- **-shared** - 1 if the group is shared, 0 if it is not.

**Examples**

- `mod group -name "mystery routers" -type device -comment "removing these devices is a bad idea, but we don't really know what purpose they serve."`
- `mod group -type device -name "border routers" -newname "defunct"
- `mod group -type device -name "border routers" -customname Location -customvalue Earth`

---

**mod ip**
Modify the properties of a ip.

**Synopsis**

```
```

**Description**

- **-ipvalue** - The ip value a.b.c.d where 0 <= a,b,c,d <= 255
- **-deviceip** - The device's ip address a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- **-ip** - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- **-host** - A valid hostname
- **-fqdn** - A valid Fully Qualified Domain Name
- **-deviceid** - A device ID
• -comment - Additional information regarding the device.
• -usetoaccess - Use this IP Value to access its device, 0 - yes, 1 - no, default - no

Examples

• mod ip -deviceip 192.0.2.10 -ipvalue 192.0.2.10 -comment "my own ip"
• mod ip -deviceip 192.0.2.10 -ipvalue 192.0.2.10 -usetoaccess 0
• mod ip -deviceid 1401 -ipvalue 192.0.2.10 -usetoaccess 0

mod metadata
Modify an existing piece of custom data associated with a specific field and associated entity.

Synopsis
mod metadata -metadataid <Metadata ID> [-fieldid <Metadata Field ID>] [-data <Data>] [-associatedtableid <Matching Row ID>]

Description

• -metadataid - ID of the custom data to delete
• -fieldid - Field ID the data is to be associated with
• -data - New data to be associated
• -associatedtableid - ID of the associated row the data corresponds to

Examples

• mod metadata -metadataid 99381 -fieldid 121 -data Room102
• mod metadata -metadataid 99381 -data Room101 -associatedtableid 21031

mod metadata field
Used to modify an existing custom data field.

Synopsis
mod metadata field -fieldid <Field ID> [-fieldname <Field Name>] [-fieldvalues <Field Values>] [-inuse <In Use>] [-flags <Allow HTML>] [-associatedtable <Associated Table>]

Description

• -fieldid - ID of the field to modify
• -fieldname - New name of the field
• -fieldvalues - List of comma separated values that the field is restricted to. If not specified, the value for this field is not restricted
• -inuse - Turns the field on or off. 1 is on, 0 is off. When the field is off, it will not be displayed with the other custom fields.
• -flags - Used for allowing HTML in the field value. 1 is allow, 0 is disallow. If disallowed, HTML will be escaped for displaying.
• -associatedtable - The table to associate this field with
Examples

- mod metadata field -fieldid 2221 -fieldname Room -fieldvalues 101,102,103,104 -inuse 1 -flags 0 -associatedtable RNDEVICE
- mod metadata field -fieldid 2221 -inuse 0

**mod module**
Modify a module's properties.

**Synopsis**
mod module -id <Module ID> [-comment <Comment>] [-customname <Customname>] [-customvalue <Customvalue>]

**Description**
- -id - The ID of a module
- -comment - Additional information about the module.
- -customname - The custom field name
- -customvalue - The custom field value

**Examples**

- mod module -id 527 -comment "Internal Use Only"

**mod partition**
Modify a partition.

**Synopsis**
mod partition -name <Name> -newname <New name> [-comment <Comment>]

**Description**
- -name - The name of the partition to be modified.
- -newname - The new name for the modified partition. Do not use this option unless you also use -name.
- -comment - Additional information regarding the partition.

**Examples**

- mod partition -name "Default Site" -newname "Redmond Site"

**mod port**
Modify a port's properties.
Synopsis
mod port -id <Port ID> [-comment <Comment>] [-customname <Customname>] [-customvalue <Customvalue>]

Description

- -id - The ID of a port
- -comment - Additional information about the port.
- -customname - The custom field name
- -customvalue - The custom field value

Examples

- mod port -id 527 -comment "Internal Use Only"

mod task
Modify a scheduled task.

Synopsis

Description

- -id - The task ID of the task to modify.
- -comment - Additional information about the task.
- -retryInterval - The number of seconds between retries.
- -expensive - Mark the task as expensive. Do not use this option with -notexpensive.
- -notexpensive - Mark the task as not expensive. Do not use this option with -expensive.
- -days - This argument differs depending on the task. For weekly tasks, -days should be a comma-separated list of weekdays. Each item in the list is a day of the week upon which the task should be run. Valid weekdays are: sun, mon, tue, wed, thu, fri, sat. For monthly tasks, -days should be a single integer between 1 and 31, corresponding to the day of the month upon which the task should be run.
- -retryCount - The number of times to retry the task if it fails.
- -repeatType - The metric by which a task repeats. Valid values are 1: once, 2: periodically, 3: daily, 4: weekly, 5: monthly. If you modify this value, then modify -repeatInterval or -days accordingly.
- -duration - Estimated duration the task will run (in minutes)
- -start - YYYY:MM:DD:HH:mm. The first date the task will run.
-repeatInterval - This option differs depending on the task. For Periodic tasks, this
is the period in minutes. For Monthly tasks, each bit of the integer (except the
last) represents a day, but we recommend using the -days option to modify the
days on which a monthly task runs. This option is invalid with all other tasks.
-approve - Approve the task
-reject - Reject the task
-override - Override the approval requirement
-customname - The custom field name
-customvalue - The custom field value

Examples
- mod task -id 7097 -repeatType 4 -days mon,wed,thur
- mod task -id 54 -retryCount 2 -duration 60
- mod task -id 54 -reject "needs technical review"

mod topology graph
Modify topology data.

Synopsis
mod topology graph -type <Topology data type> -data <Topology data value> -deviceid

Description
- type - The topology data type, typically "phy_inferred" for L1
- data - The topology data value, typically a MAC address (without colons)
- deviceid - The source device ID
- deviceportid - The source device port ID
- remotedeviceid - The destination device ID
- remotedeviceportid - The destination device port ID
- serverid - The destination server ID
- serverportid - The destination server port ID

Examples
- mod topology graph -type phy_inferred -data 0007E912C8D7 -deviceid 193 -remotedeviceid 2837
- mod topology graph -type phy_inferred -data 00123F76F759 -deviceid 193 -serverid 105001
mod unmanaged device
Modify the properties of an unmanaged device.

Synopsis
mod unmanaged device -ip <IP address> -comment <Comment>

Description
- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -comment - Additional information regarding the device.

Examples
- mod unmanaged device -ip 192.0.2.10 -comment "no need"

mod user
Modify a user's properties.

Synopsis

Description
- -u - Username
- -p - Password
- -fn - First name
- -ln - Last name
- -email - Email address
- -priv - User Privilege (1=Limited Access,2=Full Access,3=Power User,4=Admin)
- -newusername - New username for this user.
- -aaauusername - AAA username for this user.
- -aaapassword - AAA password for this user.
- -useaaaloginforproxy - Whether to use AAA logins for the Proxy Interface for this user (yes|no).
- -extauthfailover - Whether to allow external auth failover for this user (yes|no).
- -customname - The custom field name
- -customvalue - The custom field value
- -status - enable or disable

Examples
- mod user -u johnd -p new -fn Johnathan -email jdoe@example.net
- mod user -u johnd -p new -fn Johnathan -email jdoe@example.net -priv 2
mod user -u -customname Title -customValue Engineer
mod user -u johnd -status disable

**os ping**
Run a ping command from the server to the specified device.

**Synopsis**

```
os ping
```

**Description**

The ping command is an OS command. All ping options that are available at the OS level are supported. Users should be able to enter any host name or address. The behavior is that it simply passes the string to the OS, executes it as a command and returns the results of the executed command.

**Examples**

- `os ping 192.0.2.10`

**os-ping**
Run a ping command from the server to the specified device.

**Synopsis**

```
os-ping
```

**Description**

The ping command is an OS command. All ping options that are available at the OS level are supported. Users should be able to enter any host name or address. The behavior is that it simply passes the string to the OS, executes it as a command and returns the results of the executed command.

**Examples**

- `os-ping 192.0.2.10`
- `os-ping -t 192.0.2.11`

**passwd**
Change current user's password.

**Synopsis**

```
passwd -oldpwd <your old password> -newpwd <your new password>
```

**Description**

Causes the current user's password to be changed.
• -oldpwd - youroldpassword
• -newpwd - yournewpassword

Examples
• passwd -oldpwd youroldpassword -newpwd yournewpwd

pause polling
Stop polling.

Synopsis
pause polling

Description
Stop polling devices for configuration changes.

Examples
• pause polling

ping
Run a ping command on a device.

Synopsis
ping -source <IP address | Hostname | Fully Qualified Domain Name> -sourcegroup <Groupname> -dest <List of IP addresses> -rep <Task repeat period> -async -start <task start date>

Description
Causes a series of ping commands to be executed on a device. One ping command is executed for each target host specified. This series of commands may be run on the device immediately, or scheduled to run sometime in the future. Via this command, the task scheduled can be set to repeat periodically. Note that if not scheduled as a task, this command may take some time to complete.

• -source - Can be an IP address (a.b.c.d where 0 <= a,b,c,d <= 255), or a valid hostname, or a valid Fully Qualified Domain Name.
• -sourcegroup - A valid group name. Exactly one of -source or -sourcegroup must be specified.
• -dest - A comma separated list of devices. Devices may be specified in any way that is understood by the ping program on the device specified by the option "-source".
• -rep - (#min | #:# | #days | #weeks | #months) where # is a positive integer. #:# is hours:minutes, the two integers don't have to be the same. This option should not be used unless -async is also supplied.
-async - Indicates that the ping operation should be scheduled on the system as a task. The start time for the task will be immediately unless an alternate start data is provided by means of the -start option.
- start - YYYY:MM:DD:HH:mm. The date on which the task will first be run. This option should not be used unless -async is also supplied.

Examples

- ping -source 192.0.2.10 -dest 192.0.2.10
- ping -source 192.0.2.10 -dest 192.0.2.10,192.0.2.10,192.0.2.10
- ping -source 192.0.2.10 -dest 192.0.2.10 -async -start 2004:02:29:23:59 -rep 2days
- ping -source 192.0.2.10 -dest 192.0.2.10 -async
- ping -sourcegroup mygroup -dest 192.0.2.10

quit
Exit the system.

Synopsis
quit

Description
Exit the the system.

Examples

- quit

reload content

Synopsis
reload content

Description
Load any new content packs (such as scripts or policies) that have been installed on the server since the last time it was restarted or content was reloaded.

Examples

- reload content

reload drivers

Synopsis
reload drivers -force
**Description**
Causes the server to reload all installed driver files.

- `-force` - Force drivers to be reloaded even when there is error

**Examples**
- `reload drivers`
- `reload drivers -force`

---

**reload server options**

**Synopsis**
`reload server options`

**Description**
Causes the server to reload config variables from all config files.

**Examples**
- `reload server options`

---

**resume polling**

**Synopsis**
`resume polling`

**Description**
Resume polling devices for configuration changes.

**Examples**
- `resume polling`

---

**rlogin**

Make an rlogin connection to a device.

**Synopsis**
`rlogin [-override] []`

**Description**
Connect to a device through the system’s Proxy Interface via telnet (bypassing single sign-on). If you are connected to a device through a console server, you may hit ctrl-\ to return to the the system shell after logging out of the device.
- **override** - Force a connection to a device in the event that simultaneous connection warning or prevention is turned on.
- **Hostname, Device ID, Fully Qualified Domain Name, or Primary IP Address** to use to lookup the device to connect to. The characters * and ? can be used as wildcards.
- **Port** to use to connect to devices outside of the system.

**Examples**

- `rlogin 192.0.2.10`
- `rlogin -override mydevice`

---

**run advanced script**

Run an existing advanced script against a device or group of devices.

**Synopsis**


**Description**

Runs an existing advanced script, specified by name, against a device or group of devices. The proper variant of the script will be applied to each device. If no variant of the script supports a given device, that device will be skipped. The script is run as a system task.

- **-ip** - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- **-host** - A valid hostname
- **-fqdn** - A valid Fully Qualified Domain Name
- **-deviceid** - A device ID
- **-group** - A valid group name. Either a device or a group must be specified, but not both (exactly one of -ip, -hostname, -fqdn or -group must be specified).
- **-name** - Name of the advanced script to run
- **-parameters** - Command line parameters for the advanced script to run
- **-variables** - A list of variables to be replaced in the script - provided as a list of name=value pairs, separated by commas. Values can be surrounded in single-quotes ('). Within a quoted value, a single-quote can be embedded with two single-quote characters. Example: "variable1=value1,variable2='this is "value 2"'
- **-start** - YYYY:MM:DD:HH:mm. The first date on which the task will run.
- **-rep** - (#min | #:# | #days | #weeks | #months) where # is a positive integer. #:# is hours:minutes--the two integers do not have to be the same. Do not use this option with -sync.
- **-sync** - Indicates the command should return only after the snapshot retrieval task is complete. Do not use this option with -rep or -start.
-nowait - Indicates that the task should not wait if there is another task currently running against the same device.
-comment - An optional comment about the snapshot.
-presnapshot - If false, this indicates that the snapshot that runs before the script should be skipped.
-postsnapshot - If false, this indicates that the snapshot that runs after the script should be skipped. If "task", this indicates that snapshot after the script should run as a separate task.

Examples

- run advanced script -ip 192.0.2.10 -name "Extended Ping" -parameters "" -variables "Target_IP=192.0.2.10" -start 2004:02:29:23:59 -rep 2days -comment "running extended ping"
- run advanced script -group mygroup -name "Set Interface Description" -variables="interface=Ethernet1,description='provider "MCI",link id T207'" -parameters "-I /usr/etc/log.txt" -sync

run command script
Run an existing command script against a device or group of devices.

Synopsis

Description
Runs an existing command script, specified by name, against a device or group of devices. The proper variant of the script will be applied to each device. If no variant of the script supports a given device, that device will be skipped. The script is run as a system task.

- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -host - A valid hostname
- -fqdn - A valid Fully Qualified Domain Name
- -deviceid - A device ID
- -group - A valid group name. Either a device or a group must be specified, but not both (exactly one of -ip, -hostname, -fqdn or -group must be specified).
- -name - Name of the command script to run
- -variables - A list of variables to be replaced in the script - provided as a list of name=value pairs, separated by commas. Values can be surrounded in single-quotes ('). Within a quoted value, a single-quote can be embedded with two single-quote characters. Example: "variable1=value1,variable2="this is "value 2""
- -linebyline - Indicates that line by line deployment is preferred, rather than file-based deployment
-start - YYYY:MM:DD:HH:mm. The first date on which the task will run.
-rep - (#min | #:# | #days | #weeks | #months) where # is a positive integer. #:# is hours:minutes--the two integers do not have to be the same. Do not use this option with -sync.
-sync - Indicates the command should return only after the snapshot retrieval task is complete. Do not use this option with -rep or -start.
-nowait - Indicates that the task should not wait if there is another task currently running against the same device.
-comment - An optional comment about the snapshot.
-presnapshot - If false, this indicates that the snapshot that runs before the script should be skipped.
-postsnapshot - If false, this indicates that the snapshot that runs after the script should be skipped. If "task", this indicates that snapshot after the script should run as a separate task.

Examples

run command script -ip 192.0.2.10 -name "Extended Ping" -variables "Target_IP=192.0.2.10" -start 2004:02:29:23:59 -rep 2days -comment "running extended ping"
run command script -group mygroup -name "Set Interface Description" -variables="interface=Ethernet1,description='provider 'MCI',link id T207'" -linebyline -sync

run diagnostic
Run a diagnostic on a device.

Synopsis
run diagnostic [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-group <Group Name>] -diagnostic <Diagnostic Name> [-rep <Task repeat period>] [-start <Task start date>] [-sync] [-comment <Run script comment>]

Description
Run the specified diagnostic on a specified device either right away, or at some point in the future. The run diagnostic operation is actually a scheduled task.

- ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- host - A valid hostname
- fqdn - A valid Fully Qualified Domain Name
- deviceid - A device ID
- group - A name of a device group (mutually exclusive with -ip, -host, or -fqdn)
- diagnostic - A diagnostic to run. Built-in diagnostics are '@ProductAbbreviation@ Routing Table', '@ProductAbbreviation@ Interfaces' and '@ProductAbbreviation@ OSPF Neighbors'.

104  PERL API Reference Guide
-rep - (#min | #:# | #days | #weeks | #months) where # is a positive integer. #:# is hours:minutes—the two integers do not have to be the same. Do not use this option with -sync.
- start - YYYY:MM:DD:HH:mm. The first date on which the task will run. Do not use this option with -sync.
- sync - Indicates that the command should return only after the deploy task is complete. Do not use this option with -start.
- comment - An optional comment about the diagnostic.

Examples

- run diagnostic -ip 192.0.2.10 -diagnostic "vlan report" -sync
- run diagnostic -ip 192.0.2.10 -diagnostic "@ProductAbbreviation@ Routing Table" -start 2004:02:29:23:59
- run diagnostic -group "Core Routers" -diagnostic "@ProductAbbreviation@ OSPF Neighbors" -rep 7days -start 2004:01:01:01:00:00 -comment "Weekly Core Router OSPF Neighbors pull"

run external application

Execute a command.

Synopsis

run external application -app <Command> [-start <Task start date>] [-rep <Task repeat period>] [-sync] [-comment <Comment text>] [-startdir <Directory path>] [-resultfile <File path>] [-errorifnonzero <true or false>]

Description

Runs a @ProductAbbreviation@ task which spawns a new process that executes a command external to @ProductAbbreviation@.

- app - The command to execute.
- start - YYYY:MM:DD:HH:mm The time when the command will be executed. Do not use this option with -sync.
- rep - (#min | #:# | #days | #weeks | #months) where # is a positive integer. #:# is hours:minutes—the two integers do not have to be the same. Do not use this option with -sync.
- sync - Indicates that the CLI command should return only after the task is complete. Do not use this option with -start.
- comment - Comments to be attached to the task that runs to execute the command.
- startdir - The working directory of the process in which the command is executed.
- resultfile - The file to contain the output of the command.
- errorifnonzero - If true the task will be marked FAILED or WARNING if the command returns a non zero result code.
Examples

- run external application -app "grep -c /bin/csh /etc/passwd" -resultfile /home/jdoe/out.log -sync

run gc
Run the garbage collector.

Synopsis

run gc

Description
Recycle unused objects to increase the amount of free memory.

Examples

- run gc

run script
Run a command script on a device.

Synopsis


Description
Run the specified command script on a specified device either right away, or at some point in the future. The run script operation is actually a scheduled task. If no mode is specified the first supported enable, supervisor, provisioning or root mode will be used.

- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -host - A valid hostname
- -fqdn - A valid Fully Qualified Domain Name
- -deviceid - A device ID
- -group - A name of a device group (mutually exclusive with -ip, -host, or -fqdn)
- -mode - A command script mode to run the script in.
- -script - A script to run, may separate commands with \n. Commands that require multiple entries before returning to the device prompt can separate each entry with \"\r\n\".
- -rep - (#min | #::# | #days | #weeks | #months) where # is a positive integer. #::# is hours:minutes--the two integers do not have to be the same. Do not use this option with -sync.
• -start - YYYY:MM:DD:HH:mm. The first date on which the task will run. Do not use this option with -sync.
• -sync - Indicates that the command should return only after the deploy task is complete. Do not use this option with -start.
• -nowait - Indicates that the task should not wait if there is another task currently running against the same device.
• -comment - An optional comment about the script being run.

Examples

• run script -ip 192.0.2.10 -mode "Cisco IOS enable" -script "show ver" -sync
• run script -ip 192.0.2.10 -mode "Nortel BCC" -script "show system info" -start 2004:02:29:23:59
• run script -group "Core Routers" -mode "Cisco IOS configuration" -script "banner motd xCore Router\r\ndo not touch!\nprompt %h%p" -start 2004:01:01:01:00:00 -comment "Get the core routers banner and prompt standardized"

show access
Display a device access record.

Synopsis
show access -id <Device access record ID>

Description

• -id - Specifies a device access record. Think of this as a "device access record ID".

Examples

• show access -id 510

show acl
Show ACL.

Synopsis
show acl -id <Device ACL ID>

Description
Displays the device ACL in the system including Script and Application.

• -id - List only ACLs with this deviceaclid

Examples

• show acl -id 241
show basicip
Show a BasicIP model.

Synopsis
show basicip [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-id <Config ID>]

Description
If the -ip flag is given, show the BasicIP model for the most recent config for the specified device. If the -id flag is given, show the BasicIP model for the specified config. Include either the -id or -ip option, but not both.

- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -host - A valid hostname
- -fqdn - A valid Fully Qualified Domain Name
- -deviceid - A device ID
- -id - A config ID

Examples
- show basicip -ip 192.0.2.10
- show basicip -ip "East Site:192.0.2.10"
- show basicip -id 73253

show config
Show the contents of a config.

Synopsis
show config -id <Config ID> [-mask]

Description
- -id - The ID of a config
- -mask - Mask out sensitive information such as device passwords

Examples
- show config -id 2600
- show config -id 2405 -mask
show configlet
Show the configlet inbetween start and end pattern.

Synopsis
show configlet [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-start <Start Block>] [-end <End Block>] [-type <Helper>]

Description
- -ip - List all device ports on the device with this IP address
- -host - List all device ports on the device with this hostname
- -fqdn - List all device ports on the device with this Fully Qualified Domain Name
- -deviceid - List all device ports on the device with this device ID
- -start - Block start pattern for the configlet
- -end - Block end pattern for the configlet
- -type - Type (helper) of the configlet

Examples
- show configlet -ip 1.2.3.4 -start webfarm -type C_POOL

show device
Show a device's properties.

Synopsis
show device [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-id <Device ID>]

Description
- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -host - A valid hostname
- -fqdn - A valid Fully Qualified Domain Name
- -deviceid - A device ID
- -id - A device ID

Examples
- show device -ip 192.0.2.10
- show device -ip "East Site:192.0.2.10"
- show device -id 527
show device config
Show the config most recently retrieved from the specified device.

Synopsis
show device config [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>]

Description

• -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
• -host - A valid hostname
• -fqdn - A valid Fully Qualified Domain Name
• -deviceid - A device ID

Examples

• show device config -ip 192.0.2.10
• show device config -ip "East Site:192.0.2.10"

show device family
Show the family classification associated with the specified device.

Synopsis
show device family -ip <IP address>

Description

• -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.

Examples

• show device family -ip 192.0.2.10
• show device family -ip "East Site:192.0.2.10"

show device latest diff
Show the difference between two configs most recently retrieved from the specified device.

Synopsis
show device latest diff [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>]
show deviceinfo
Show a DeviceInformation model.

Synopsis
show deviceinfo [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-id <Config ID>]

Description
If the -ip flag is given, show the DeviceInformation model for the most recent config for the specified device. If the -id flag is given, show the DeviceInformation model for the specified config. Include either the -id or -ip option, but not both.

Examples
• show deviceinfo -ip 192.0.2.10
• show deviceinfo -ip "East Site:192.0.2.10"
• show deviceinfo -id 73253

show diagnostic
Show a diagnostic's results.

Synopsis
show diagnostic -id <Diagnostic ID>
Description

- id - A diagnostic ID

Examples

- show diagnostic -id 73253

show driver
Show the driver assigned to a device.

Synopsis

show driver [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>]

Description

- ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- host - A valid hostname
- fqdn - A valid Fully Qualified Domain Name
- deviceid - A device ID

Examples

- show driver -ip 192.0.2.10
- show driver -host rtr5.vfm.lab

show event
Display the details of an event.

Synopsis

show event -id <event ID>

Description

- id - A valid event id

Examples

- show event -id 27

show group
Show all information for a group.
Synopsis
show group [-name <Group name>] [-id <Group id>]

Description

- -name - The group name for whom information will be displayed
- -id - The group id for whom information will be displayed

Examples

- show group -name johnd
- show group -id 5

show icmp
Show an ICMPTest model.

Synopsis
show icmp [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-id <Config ID>]

Description
If the -ip flag is given, show the ICMPTest model for the most recent config for the specified device. If the -id flag is given, show the ICMPTest model for the specified config. Include exactly one of the -id or -ip option.

- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -host - A valid hostname
- -fqdn - A valid Fully Qualified Domain Name
- -deviceid - A device ID
- -id - A config ID

Examples

- show icmp -ip 192.0.2.10
- show icmp -id 73253

show int
Show a ShowInterfaces model.

Synopsis
show int [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-id <Config ID>]

PERL API Reference Guide
Description
If the -ip flag is given, show the ShowInterfaces model for the most recent config for the specified device. If the -id flag is given, show the ShowInterfaces model for the specified config. Include either the -id or -ip option, but not both.

- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -host - A valid hostname
- -fqdn - A valid Fully Qualified Domain Name
- -deviceid - A device ID
- -id - A config ID

Examples

- show int -ip 192.0.2.10
- show int -ip "East Site:192.0.2.10"
- show int -id 73253

show ip
Show a ip's properties.

Synopsis
show ip -ipvalue <Value> [-deviceip <Device IP address>] [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>]

Description

- -ipvalue - The ip value a.b.c.d where 0 <= a,b,c,d <= 255
- -deviceip - The device's ip address a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -host - A valid hostname
- -fqdn - A valid Fully Qualified Domain Name
- -deviceid - A device ID

Examples

- show ip -deviceip 192.0.2.10 -ipvalue 192.0.2.10
- show ip -deviceid 1401 -ipvalue 192.0.2.10
show latest access
Show the most recent access record for the specified device.

Synopsis
show latest access [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>]

Description
- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -host - A valid hostname
- -fqdn - A valid Fully Qualified Domain Name
- -deviceid - A device ID

Examples
- show latest access -ip 192.0.2.10
- show latest access -ip "East Office:192.0.2.10"

show metadata
Show a specific piece of custom data.

Synopsis
show metadata -metadataid <Metadata ID>

Description
- -metadataid - ID of the custom data to show

Examples
- show metadata -metadataid 54535

show metadata field
Show a custom data field

Synopsis
show metadata field -fieldid <Field ID>

Description
- -fieldid - ID of the custom data field to show

Examples
- show metadata field -fieldid 8394
show module
Show a module's properties.

Synopsis
show module -id <Module ID>

Description
- -id - The ID of a module

Examples
- show module -id 527

show ospfneighbor
Show a ShowOSPFNeighbors model.

Synopsis
show ospfneighbor [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-id <Config ID>]

Description
If the -ip flag is provided, show the ShowOSPFNeighbors model for the most recent config for the specified device. If the -id flag is given, show the ShowOSPFNeighbors model for the specified config. Include either the -id or -ip option, but not both.

- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -host - A valid hostname
- -fqdn - A valid Fully Qualified Domain Name
- -deviceid - A device ID
- -id - A config ID

Examples
- show ospfneighbor -ip 192.0.2.10
- show ospfneighbor -ip "East Site:192.0.2.10"
- show ospfneighbor -id 73253

show permission
Display whether or not a user has permissions for a particular resource.

Synopsis
show permission -resource <resource> [-u <username>] [-id <user ID>]


Description

- -resource - The name of a Command or Resource.
- -u - Username
- -id - User ID

Examples

- show permission -resource com.rendition.dib.DeleteACLTask -u bob
- show permission -resource "Add Device" -id 101715

show policy
Shows policy information

Synopsis
show policy -id <Policy ID>

Description

- -id - policy id

Examples

- show policy -id 6120

show policy compliance
Shows policies and device compliance states

Synopsis
show policy compliance [-policyid <Policy ID>] [-deviceid <Device ID>] [-compliance <Compliance State (in|out|unknown)>]

Description

- -policyid - policy id
- -deviceid - device id
- -compliance - compliance state (in|out|unknown)

Examples

- show policy compliance
- show policy compliance -policyid 6120
- show policy compliance -deviceid 312
- show policy compliance -policyid 6120 -deviceid 312
- show policy compliance -policyid 6120 -compliance in
- show policy compliance -deviceid 25549 -compliance out
**show policy rule**
Shows rule information

**Synopsis**
show policy rule -id <Rule ID>

**Description**
- -id - rule id

**Examples**
- show policy rule -id 3508

**show polling status**
Show the current status of polling.

**Synopsis**
show polling status

**Description**

**Examples**
- show polling status

**show port**
Show a port's properties.

**Synopsis**
show port -id <Port ID>

**Description**
- -id - The ID of a port

**Examples**
- show port -id 527

**show routing**
Display a routing table.
Synopsis
show routing [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-id <Routing table ID>]

Description
If the -ip flag is given, show the most recent routing table captured for the specified device. If the -id flag is given, show the specified routing table. Include either the -id or -ip option, but not both.

- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -host - a valid hostname
- -fqdn - A valid Fully Qualified Domain Name
- -deviceid - A device ID
- -id - A routing table ID

Examples
- show routing -host rtr6.vfm.lab
- show routing -id 3276

show rule condition
Shows rule condition information

Synopsis
show rule condition -id <Condition ID>

Description
- -id - condition id

Examples
- show rule condition -id 3508

show script
Show one command script, advanced script or diagnostic.

Synopsis
show script [-id <Script / Diagnostic ID>] [-name <Script / Diagnostic Name>] [-type <Script / Diagnostic Type>]

PERL API Reference Guide  119
Description
Output the indicated command script, advanced script or diagnostic. The desired script or diagnostic can be specified by ID, or by a combination of name and type. If more than one name match occurs, then an error will be reported and you must specify the unique script desired by ID.

- **-id** - ID of the desired script or diagnostic
- **-name** - Name of the desired script or diagnostic
- **-type** - Type of the desired script or diagnostic - may be command, advanced or diagnostic

Examples

- show script -id 5
- show script -name "Edit Port Duplex" -type command

---

**show server option**
Display the setting of a server option

**Synopsis**

```
show server option -name <option name> [-default <default value>]
```

**Description**
Display the value of an Admin Setting or server configuration option. If the option is not set and no default is provided then this command will fail.

- **-name** - The name of the server option.
- **-default** - The value to return if the option is not set.

**Examples**

- show server option -name proxy/ssh_listener_port
- show server option -name customer/https_port -default 443

---

**show session**
Show interceptor log record.

**Synopsis**

```
show session -id <Interceptor log id>
```

**Description**

- **-id** - Interceptor log ID
Examples

- show session -id 5

**show session commands**
List all commands in interceptor log record.

**Synopsis**
show session commands -id <Interceptor log id>

**Description**

- -id - Interceptor log ID

**Examples**

- show session commands -id 5

**show snapshot**
Show the config most recently retrieved from the specified device.

**Synopsis**
show snapshot [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>]

**Description**

- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -host - A valid hostname
- -fqdn - A valid Fully Qualified Domain Name
- -deviceid - A device ID

**Examples**

- show snapshot -ip 192.0.2.10
- show snapshot -ip "East Site:192.0.2.10"

**show system message**
Display the details of a system message.

**Synopsis**
show system message -id <System message ID>
Description

- -id - A valid system message id

Examples

- show system message -id 27

---

show task
Shows detailed information about a task.

Synopsis

show task -id <Task ID>

Description

- -id - The task ID whose details will be displayed

Examples

- show task -id 354

---

show topology
Show details for a single topology record.

Synopsis

show topology -id <Topology Data ID>

Description

- -id - The id of the topology record to show.

Examples

- show topology -id 6543201

---

show user
Show all information for a user.

Synopsis

show user [-u <User name>] [-id <User id>]

Description

- -u - The user name for whom information will be displayed
- **-id** - The user id for whom information will be displayed

### Examples

- show user -u johnd
- show user -id 5

---

#### show version

**Synopsis**

show version

**Description**

Display the release version of @ProductAbbreviation@.

**Examples**

- show version

---

#### source

Have the system client execute all commands contained within a text file.

**Synopsis**

source <The name of the file containing CLI commands to execute.>

**Description**

This command has no options but takes one argument: the name of the file to "source". The source file should contain only valid CLI commands each separated by one newline.

**Examples**

- source C:\temp\commands.txt

---

#### ssh

Make an ssh connection to a device.

**Synopsis**

ssh [-override] []

**Description**

Connect to a device through the system’s Proxy Interface via ssh (bypassing single sign-on). If you are connected to a device through a console server, you may hit ctrl-\ to return to the system shell after logging out of the device.

- **-override** - Force a connection to a device in the event that simultaneous connection warning or prevention is turned on.
- Hostname, Device ID, Fully Qualified Domain Name, or Primary IP Address to use to lookup the device to connect to. The characters * and ? can be used as wildcards.
- Port to use to connect to devices outside of the system.

Examples

- ssh 192.0.2.10
- ssh -override mydevice

stop task
Stop a running task.

Synopsis
stop task -id <Task ID>

Description

- -id - The task ID of the task to stop.

Examples

- stop task -id 54

stop task all
Stop all Running and Waiting tasks.

Synopsis
stop task all

Description

Examples

- stop task all

synchronize
Synchronize a device's startup and running configs.

Synopsis

124  PERL API Reference Guide
Description
Synchronize a device's startup configuration so it matches its running configuration. The synchronize operation is actually a scheduled task.

- -ip - a.b.c.d where 0 <= a,b,c,d <= 255. You may optionally prefix the IP with SITE: where SITE is the name of the Site the device is in.
- -host - A valid hostname
- -fqdn - A valid Fully Qualified Domain Name
- -deviceid - A device ID
- -group - A name of a device group (mutually exclusive with -ip, -host, or -fqdn)
- -skipinsync - Indicates that the command should skip any device that the system indicates already has matching startup and running configs.
- -rep - (#min | #:# | #days | #weeks | #months) where # is a positive integer. #:# is hours:minutes--the two integers do not have to be the same. Do not use this option with -sync.
- -start - YYYY:MM:DD:HH:mm. The first date on which the task will run. Do not use this option with -sync.
- -sync - Indicates that the command should return only after the synchronize task is complete. Do not use this option with -start.
- -comment - An optional comment about the synchronize task.

Examples
- synchronize -ip 192.0.2.10 -sync
- synchronize -ip 192.0.2.10 -start 2004:02:29:23:59
- synchronize -group "Core Routers" -skipinsync -start 2004:01:01:01:00:00 -comment "Make sure core routers have matching startup and running configs"

telnet
Make a telnet connection to a device.

Synopsis
telnet [-override] []

Description
Connect to a device through the system's Proxy Interface via telnet (bypassing single sign-on). If you are connected to a device through a console server, you may hit ctrl-\ to return to the the system shell after logging out of the device.

- -override - Force a connection to a device in the event that simultaneous connection warning or prevention is turned on.
- - Hostname, Device ID, Fully Qualified Domain Name, or Primary IP Address to use to lookup the device to connect to. The characters * and ? can be used as wildcards.
- - Port to use to connect to devices outside of the system.
Examples

- telnet 192.0.2.10
- telnet -override mydevice

---

test config

Test policy compliance for a device configuration script.

Synopsis

test config -family <Device Family> -script <Configuration Script> [-policy <Policy Name>] [-group <Device Group>]

Description

This command is used to verify whether a configuration script is in compliance with applicable policies.

- -family - The device family for the configuration script to be tested ("Cisco IOS", F5, etc.)
- -script - The configuration script to be tested.
- -policy - The name of the policy for which the script will be tested against.
- -group - Specify a device group name. The test will be performed against the policies that are applicable to the group. If both -policy and -group are used, -group argument will be ignored. If none of -policy and -group is used, test will be performed against all applicable policies.

Examples

- test config -family "Cisco IOS" -script "version 12.1 ...."
- * Note this command is intended for API use since it is difficult to input the entire configuration script in the command line.

---

test software

Test software compliance for a device or device group.

Synopsis

test software [-ip <IP Address>] [-group <Device Group>]

Description

- -ip - The IP address of a single device to test.
- -group - A device group containing multiple devices to test.

Examples

- test software -ip 192.0.2.10
- test software -group CoreRouters
**traceroute**

Run a traceroute command on a device.

**Synopsis**

traceroute -source <IP address | Hostname | Fully Qualified Domain Name> -
sourcegroup <Group name> -dest <List of devices> -rep <Task repeat period> -async -
start <task start date>

**Description**

Causes a series of traceroute commands to be executed on a device. One traceroute command is executed for each target host specified. This series of commands may by run on the device immediately, or scheduled to run sometime in the future. Via this command, the task scheduled can be set to repeat periodically. Note that if not scheduled as a task, this command may take some time to complete.

- **-source** - Can be an IP address (a.b.c.d where 0 <= a,b,c,d <= 255), or a valid hostname, a valid Fully Qualified Domain Name.
- **-sourcegroup** - A valid group name. Exactly one of -source or -sourcegroup must be specified.
- **-dest** - A comma seperated list of devices. Devices may be specified in any way that is understood by the traceroute program on the device specified by the option "-source".
- **-rep** - (#min | #:# | #days | #weeks | #months) where # is a positive integer. #:# is hours:minutes, the two integers don't have to be the same. This option should not be used unless -async is also supplied.
- **-async** - Indicates that the traceroute operation should be scheduled on the system as a task. The start time for the task will be immediatly unless an alternate start data is provided by means of the -start option.
- **-start** - YYYY:MM:DD:HH:mm. The date on which the task will first be run. This option should not be used unless -async is also supplied.

**Examples**

- traceroute -source 192.0.2.10 -dest 192.0.2.10
- traceroute -source 192.0.2.10 -dest 192.0.2.10,192.0.2.10,192.0.2.10
- traceroute -source 192.0.2.10 -dest 192.0.2.10 -start 2004:02:29:23:59 -rep 2days
- traceroute -source 192.0.2.10 -dest 192.0.2.10 -async
- traceroute -sourcegroup mygroup -dest 192.0.2.10
**undeploy image**

undeploy software images from device

**Synopsis**

undeploy image -ip <device ip address> -images <images separated by ,> [-reboot <reboot instruction>] [-rebootwait <reboot wait (in seconds)>] [-filesystem <file system of device>] [-pretask <task to run before delete>] [-posttask <task to run after delete>]

**Description**

delete software images from device.

- -ip - ip address of the device the images will be deleted.
- -images - images to be deleted.
- -reboot - wheather to reboot the device after deleting images.
- -rebootwait - seconds to wait before reboot.
- -filesystem - name of filesystem of the device the images will be deleted.
- -pretask - name of task before delete.
- -posttask - name of task after delete.

**Examples**

- undelete image -ip 10.1.1.1 -images bar.bin,baz.bin -filesystem flash:
- undelete image -ip 10.1.1.1 -images bar.bin,baz.bin -filesystem flash: -reboot -rebootwait 60
- undelete image -ip 10.1.1.1 -images bar.bin,baz.bin -filesystem flash: -reboot -rebootwait 60 -posttask squeeze

**update dynamic group**

Update dynamic group's member devices.

**Synopsis**

update dynamic group -name <Group name>

**Description**

Recalculate a dynamic group's member devices based on the predefined criteria. This has no effect on a non-dynamic device group.

- -name - The group name for which the member devices will be updated

**Examples**

- update dynamic group -name "all device out of compliance"
version
Display the system version.

Synopsis
version

Description
Displays the system version build number.

Examples
- version