# Java API Reference for Network Compliance Manager 1.3

CiscoWorks

**Corporate Headquarters**
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

Text Part Number: OL-10253-06

*Java API Reference Guide for Network Compliance Manager 1.3*

# Table of Contents


Getting Started ..... 5
Intended Audience ..... 5
Document Conventions ..... 5
Requirements ..... 5
NCM ..... 5
License ..... 5
Operating Systems ..... 6
Java ..... 6
Overview ..... 7
Why integrate? ..... 7
Why Java? ..... 7
Programming Model ..... 7
Centralized or Distributed Applications ..... 7
Request/Response ..... 7
Threading model ..... 8
Relationship to JDBC ..... 8
Windows Installation ..... 9
Installing from CD ..... 9
LibraryJARs ..... 9
NCM API JAR ..... 9
Configuration Files ..... 9
Samples ..... 9
Documentation ..... 9
Setting Up a Command-line Environment ..... 10
Setting up an integrated development environment ..... 10
Unix Installation ..... 11
Installing from CD ..... 11
LibraryJARs ..... 11
NCM API JAR ..... 11
Configuration Files ..... 11
Samples ..... 11
Documentation ..... 12
Setting Up a Command-line Environment ..... 12

Setting up an integrated development environment ..... 12
Programming with the NCM Java API ..... 13
Working with the Session object ..... 13
Session contexts ..... 13
UserIDs and Permissions ..... 13
Executing requests ..... 13
Relationship between the API and the CLI or Telnet/SSH Proxy ..... 13
Handling results ..... 14
Status ..... 14
Simple results ..... 14
Complex results: ResultSet type ..... 14
Exceptions ..... 14
Metadata ..... 15
Integration Hooks ..... 16
Run External Application tasks ..... 16
Callbacks ..... 16
Approver callback interface ..... 16
Approver use cases ..... 17
Approver coding ..... 17
Cleaner callback interface ..... 17
Cleaner use case ..... 18
Cleaner coding ..... 18
Commands ..... 19
Permissions ..... 19
Commands and Return Values ..... 19
ResultSet Contents ..... 22
Permissions ..... 27
Appendix A: NCM Documentation ..... 29
Appendix B: Obtaining Documentation, Obtaining Support, and Security Guidelines ..... 30
Appendix C: CLI/API Command Reference ..... 31

# Getting Started

## Intended Audience

This document is intended for network engineering who:

- Write scripts to automate device configuration.
- Are comfortable with basic Java programming, and have an understanding of database schema and access methods.
- Have knowledge of the CiscoWorks Network Compliance Manager (NCM) Command Line Interface (CLI). CLI documentation is available in Appendix A of the *User Guide for Network Compliance Manager 1.3*, or can be accessed within the CLI using the “help” command. Information is also available through the Java API.
- Integrate various third party systems with NCM 1.3, such as network management, workflow, and trouble ticketing solutions.

## Document Conventions

This document uses the following conventions:

- File names, directory names, and answers/arguments supplied by the user are represented in italics. For example: *NCMAPI.zip*
- Display of on-screen activity is represented in Courier font. For example:
  `Volume Serial Number`

# Requirements

The following are required to use the NCM Java API.

## NCM

To use this version of the NCM Java API, you must be running NCM 1.3. The server must be running and accessible to the client where your application runs via port 1099 (Java API).

A copy of the NCM client package is required to write the programs and run the examples. The NCM client package is available as part of the NCM distribution.

**Note:** The server can be bound to a port other than 1099, but in this case, the session API must be explicitly provided with the port number.

## License

You must have a valid license on the NCM server to use the NCM Java API.

### Operating Systems

The NCM Java 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
- Red Hat Linux Enterprise AS (update 2 and 3)
- Solaris 9.x

### Java

You will need to download the Java JDK from Sun. The JDK can be downloaded from *http://java.sun.com/downloads/*

The NCM Java API is tested with Java version 1.4.2.

# Overview

NCM is a powerful software solution for network configuration control with sophisticated Web and command-line interfaces for interactive use with NCM. Java Application Programming Interface (API) adds another dimension to NCM by integrating NCM with other software. You can link NCM to a variety of third-party and custom-built applications, such as ticketing, asset tracking, workflow, change request, and network management software solutions.

## Why integrate?

When a ticketing or NMS product is used side-by-side with NCM, the two products are good, but when they interact with each other, they are even better. By using the Java API to integrate NCM with third party software, you can multiply the value provided by both products. This means fewer errors, faster turnaround and improved day-to-day efficiency.

## Why Java?

Java is a modern, object-oriented language that can run on a variety of platforms. It lends itself to high performance, scalable, highly available solutions. Java applications are also flexible and highly maintainable. Java is the right choice when you have professional development resources, performance is important, and your solution is expected to be in use for the long term.

## Programming Model

The NCM Java API is designed to expose a straightforward programming model with a relatively small set of objects to learn.

### Centralized or Distributed Applications

The NCM Java API can be accessed from your NCM server machine. This is the simplest programming model.

The NCM Java API also enables you to run applications remotely from the NCM server. This means you can run NCM on machine A and API-based applications on machine B. In distributed software terminology, this is called *remoting*. By remoting your application, you will create a client/server application where NCM is the server and your application is the client. This might be desirable for load balancing, ease of setting up a development environment, security, or a variety of other reasons.

If you use remoting, you will need your network configured to allow traffic on port 1099 (Java RMI) to reach the NCM server.

### Request/Response

The NCM Java API generally follows a request/response model. Your application makes a request via the `exec` method and waits for a response from the server. Details are explained in the “Programming with the NCM Java API” section.

### Threading model

The NCM Java API is synchronous on the client side and asynchronous on the server side. This means that when your application makes a request, the calling thread in your application is blocked until a response is available from the server.

This response may mean that your command has been executed and the results returned (such as `list user` command, which returns a list of users immediately), or it may mean that an NCM task has been created and queued for future execution (such as the `get snapshot` command, which will schedule a NCM snapshot task).

If you want to issue multiple overlapped commands, you will need to use standard Java multi-threading techniques in your application.

### Relationship to JDBC

You may notice a strong similarity between the NCM Java API and certain JDBC calls. In particular, NCM returns results in a `ResultSet` object derived from JDBC. This makes it easier for developers familiar with JDBC to get up and running with the NCM API. Some of the `ResultSet` methods are not applicable to NCM, and will return exceptions if used. These are detailed in the “Programming with the NCM Java API” section.

# Windows Installation

## Installing from CD

This section describes how to install all the necessary components of the NCM Java API from the installer CD.

From the NCM installation CD, use either the Server or Client installation options. With either of these options, you will get a copy of:

- The Java Runtime Environment
- The NCM client JAR
- Any library JAR files necessary to run the NCM SDK

If you are running the SDK application on a machine that already has a NCM server installed, no further installation is required. Use the Client install if you will be connecting to a remote NCM server for your SDK application.

**Note**: The file paths referenced in this document assume you installed NCM to the default file location, `c:\rendition`. If you installed NCM to a different location, then replace `c:\rendition` with the root directory that you provided at install time.

### LibraryJARs

Library JARs are located at `c:\rendition\jre\lib\ext`. If you do not run your SDK application with the JRE, then you must set your classpath accordingly.

### NCM API JAR

The NCM API JAR is located at `c:\rendition\client\truecontrol-client.jar`. It may be moved to another location.

## Configuration Files

NCM and the NCM Java API use several configuration files with an RCX extension. If you use the NCM Java API on the same machine that you installed NCM to, the RCX files will already be where they need to be, in the directory `c:\rendition\jre`. If you want to use the API on another machine, you need to manually copy the RCX files to the JRE/JDK directory on the other machine.

The RCX configuration files used by the NCM API are:

- messages.rcx
- logging.rcx
- commandlineclient.rcx

## Samples

The API samples are located in `c:\rendition\client\sdk\examples\java`.

## Documentation

The NCM Java API documentation consists of the Javadocs for the API. Javadocs are located in `<install directory>\client\sdk\docs\api.`

## Setting Up a Command-line Environment

If you are invoking javac and java from the command line, you can easily set up a command line environment to prepare to use the NCM API. Append `truecontrol-client.jar` to your classpath.

> **Note**: Do not put the `truecontrol-client.jar` in jre/lib/ext. NCM's Java processes will not start.

Example:

```
set
CLASSPATH=%CLASSPATH%;.;c:\src\java\myproject\classes;c:\rendition\clie
nt\truecontrol-client.jar
```

To verify that your environment is correct, please compile and run `Example0.java`. Here is what you should see:

```
C:\Rendition\client\sdk\examples\java>set CLASSPATH=.;c:\rendition\client\
truecontrol-client.jar

C:\Rendition\client\sdk\examples\java>javac -d . Example0.java

C:\Rendition\client\sdk\examples\java>java com.rendition.api.examples.Exam
ple0
Starting Example0
Session connectivity verified

C:\Rendition\client\sdk\examples\java>
```

## Setting up an integrated development environment

Setting up for an IDE is similar to the command-line environment. You need to provide the location of `truecontrol-client.jar` to your IDE. In many editors, this is an option for the project. Details follow for selected IDEs.

**JCreator Pro:**

1. Go to the menu Project:Project Settings.
2. In the project settings dialog box, go to the Required Libraries tab and select New…
3. Enter the name "truecontrolAPI".
4. Click the Add button and then select Add Archive from the popup menu.
5. Navigate to the correct directory and select `truecontrol-client.jar`.

**Jbuilder 5:**

1. Go to the menu Project:Project Properties.
2. In the dialog box, select the Paths tab then the Required Libraries sub-tab.
3. Click Add and then the New button.
4. Enter the name "truecontrolAPI."
5. Navigate to the correct directory and select `truecontrol-client.jar`

# Unix Installation

## Installing from CD

This section describes how to install all the necessary components of the NCM Java API from the installer CD.

From the NCM installation CD, use either the Server or Client installation options. With either of these options, you will get a copy of:

- The Java Runtime Environment
- The NCM client JAR
- Any library JAR files necessary to run the NCM SDK

If you are running the SDK application on a machine that already has an NCM server installed, no further installation is required. Use the Client install if you will be connecting to a remote NCM server for your SDK application.

**Note**: The file paths referenced in this document assume you installed NCM to the default file location, `<install directory>/jre`. If you installed NCM to a different location, then replace `<install directory>/jre` with the root directory that you provided at install time.

### LibraryJARs

Library JARs are located at `<install directory>/jre/lib/ext`. If you do not run your SDK application with the JRE, then you must set your classpath accordingly.

### NCM API JAR

The NCM API JAR is located at `<install directory>/jre/client/truecontrol-client.jar`. It may be moved to another location.

## Configuration Files

NCM and the NCM Java API use several configuration files with an RCX extension. If you use the NCM Java API on the same machine that you installed NCM to, the RCX files will already be where they need to be, in the directory `<installed directory>/jre`. If you want to use the API on another machine, you need to manually copy the RCX files to the JRE/JDK directory on the other machine.

The RCX configuration files used by the NCM API are:

- messages.rcx
- logging.rcx
- commandlineclient.rcx

## Samples

The API samples are located in `/usr/local/client/sdk/examples/java`

## Documentation

The NCM Java API documentation consists of the Javadocs for the API. Javadocs are located in `<install directory>/client/sdk/docs/api`.

## Setting Up a Command-line Environment

If you are invoking javac and java from the command line, you can easily set up a command line environment to prepare to use the NCM API. Append `truecontrol-client.jar` to your classpath.

Do not put the `truecontrol-client.jar` in jre/lib/ext. NCM' Java processes will not start.

To verify that your environment is correct, compile and run `Example0.java`. Here is what you should see:

```
bash# export CLASSPATH= $CLASSPATH :.:/<install
directory>/client/truecontrol-Client.jar

bash# <install directory>/jre/bin/javac -d . Example0.java

bash# <install directory>/jre/bin/java com.rendition.api.examples.Exam
ple0

Starting Example0
Session connectivity verified

<install directory>/client/sdk/examples/java>
```

## Setting up an integrated development environment

Setting up for an IDE is similar to the command-line environment. You need to provide the location of `truecontrol-client.jar` to your IDE. In many editors, this is an option for the project. Details follow for selected IDEs.

**JCreator Pro:**

1. Go to the menu Project:Project Settings.
2. In the project settings dialog box, go to the Required Libraries tab and select New…
3. Enter the name "truecontrolAPI."
4. Click the Add button then select Add Archive from the popup menu.
5. Navigate to the correct directory and select `truecontrol-client.jar`.

**Jbuilder 5:**

1. Go to the menu Project:Project Properties.
2. In the dialog box, select the Paths tab then the Required Libraries sub-tab.
3. Click Add then the New button.
4. Enter the name "truecontrolAPI."
5. Navigate to the correct directory and select `truecontrol-client.jar`

# Programming with the NCM Java API

## Working with the Session object

All interaction with the NCM Java API starts with a Session object.

### Session contexts

`Session.open` creates a session context for execution of commands. This method actually contacts the NCM server via Java RMI on port 1099, and authenticates the user using the supplied arguments. The server parameter is optional; if omitted, localhost will be contacted.

Make sure that you close the session context when done with it via the `Session.close` method. Like file handles, there is a finite supply of sessions.

Session objects are thread-safe, so you may use the Session object across threads to do overlapping operations.

### UserIDs and Permissions

When opening the session, you must provide a user name and password for a valid NCM user. NCM makes no distinction between the user identities used to log into the WebUI, CLI, or Telnet/SSH Proxy and those used to access the API.

Each NCM API call will be validated against the user identity provided to ensure the user has sufficient privileges to run the requested operation, just as the user's privileges would be validated by the WebUI, CLI, or Telnet/SSH Proxy.

We suggest that you set up dedicated NCM users for API access, with appropriate privilege levels for the kinds of applications you are writing. For example, an application that only retrieves data from NCM might require a Limited Access user, whereas an application to remove out-of-date information from the system would require Admin privileges.

When calling `Session.open`, note that the user name and password are case sensitive. If you provide bad authentication information, you will receive a CiscoAPIException.

## Executing requests

You can send commands to the NCM server through the Session object.

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

`Session.exec` is used to send a request to the NCM 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 by telnetting 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 API does not support these.

## Handling results

This section covers the returned objects and exceptions thrown by `Session.exec`.

### Status

The return value from `Session.exec` is a Return object. `Return.getSucceeded` will return `true` if the command completed successfully; or `false if the command failed`. You can get extended information codes via `Return.getReturnStatus.` The status codes vary based on the type of request; they are documented in the Commands section.

Certain API commands are processed by the NCM server asynchronously. In these cases, the return value only indicates that the command was accepted without errors. The final result of executing the command must be determined by waiting until the corresponding NCM task has completed and inspecting the task results. The commands that are processed asynchronously are indicated by a checkmark in the appropriate column of the table in the Commands section.

### Simple results

If the command returns a simple String result, use Return.getString to examine the result. The commands with String results are shown in the Commands section.

### Complex results: ResultSet type

Many commands return a complex result with many fields, or several rows of such field-based data. The commands with complex results are shown below in the Commands section.

The NCM API uses JDBC's ResultSet interface to provide access to complex results. You can learn more about this interface in numerous books and online resources for JDBC. The samples `Example1.java, Example2.java` and `Example3.java` all show how to work with ResultSet data.

To interact with ResultSet data, you must know the valid columns and types for each command. This information is provided below in the Commands section, under the table heading Return Value(s). You can also use the metadata interface to work with ResultSets in a generic way, so that you do not have to hard code the data types being returned from a given command.

### Exceptions

The following exceptions are sent by Session.exec. Details can be found in the javadocs.

- CiscoAPIException: Generic API exceptions.
- ResultSetException: Thrown when incorrect method is used to retrieve a field from a ResultSet, e.g. calling getInt on a String field.
- NotSupportedException: Thrown when an unsupported ResultSet method is called. See the javadocs to determine which methods are supported.

## Metadata

Metadata (meaning data-about-data) describes the data fields returned in a ResultSet. This can be used to determine how many fields were returned in the result set, the name for each field, and the data type for each field. `ResultSet.getMetaData` is the method that returns metadata for a result set.

`Example3.java` shows a useful application for metadata, processing any user-supplied command. You can see how metadata is required to print results from a command whose identity is not known at compile time.

**Note:** *Developers familiar with C-based languages such as Java and C++ should take special note: the column indexes for all metadata methods are 1-based not 0-based.*

## Integration Hooks

### Run External Application tasks

NCM' Run External Application task enables you to invoke applications and scripts from within NCM. This includes the ability to run your own NCM API applications. In other words, you can extend NCM' functionality by using this API to write your own application that integrates with outside applications and datasources.

Using the Web UI, you can configure NCM to invoke your own application when certain system events occur. Note that if you need to call out to third party software from your custom application, you have several options:

- Use that application's Java API, if one is provided.
- Use that application's non-Java API via RMI.
- Use a communication channel such as message queuing, CORBA, sockets, and so on.
- Interact via the file system or databases.
- Call that application directly via `Runtime.getRuntime().exec()`

### Callbacks

There are two important callback methods from NCM to your Java code that you can use to customize the NCM engine:

- The Approver interface
- The Cleaner interface

Note that these callbacks cannot be remoted. The code must be present on the NCM server. If desired, you can provide a server-side stub which uses your own RMI calls to pass the call along to the client.

Also note that the following directions require you to modify NCM' configuration files. Make sure to keep a backup copy, as a corrupted configuration may make the server unstable.

### Approver callback interface

The approver interface is provided to allow an external ticketing system to approve or deny a particular user's access to a device.

NCM will call the user-provided approver in the following circumstances:

- Before the Telnet/SSH Proxy opens a device session – approveInterceptorSessionLogin() is invoked
- Before a device configuration is modified – hasModifyConfigPermission() is invoked
- Before a device group configuration is modified – hasGroupModifyConfigPermission() is invoked
- Before any CLI command is processed – hasPermission() is invoked

See the javadoc comments for details on when these methods are invoked, and what parameters are passed. Note, some methods are overloaded.

## Approver use cases

Here are two possible cases where this might be useful. The cases posit integration of NCM with a third-party ticketing system (3PT).

**Case 1: External task approval**

- *Network Engineer* – Schedules a config deployment for ticket T and work request W.
- *NCM* – Requests approval for change to device D with ticket T and work request W.
- *Ticketing System* – Returns true or false with a reason R
- *NCM* – Lets the task run, or marks it as failed setting the Result to 'not approved by 3PT because R'

Third party ticketing system (3PT) should return true or false synchronously using internal data (such as time of day and ticket status) so no timeout is needed.

**Case 2: External session approval**

- *Network Engineer* – Requests session on Device D for work request W.
- *NCM* – Requests approval for connection to device D for work request W
- *Ticketing System* – Returns true or false with a reason R
- *NCM* – Starts the session or displays the error 'Session not approved by 3PT because R'

## Approver coding

NCM will use the configuration file appserver.rcx to determine what class to use for the session approver. A default do-nothing (always approve) approver, `com.rendition.api.DefaultApprover`, is provided by NCM.

To install your own approver, follow these steps:

- Code your own approver that implements the `com.rendition.api.Approver` interface
- Modify "approver/className" option in appserver.rcx file, specifying your own class.
- Build a JAR file that contains all your new classes and copy it into %JBOSS_HOME%/server/default/lib directory.

## Cleaner callback interface

The cleaner interface is provided to allow custom actions upon user exiting a NCM device session. NCM will call the user-provided cleaner when the Telnet/SSH Proxy closes a device session.

## Cleaner use case

### Case 1: External change annotation

- *Network Engineer* – Configures Device D for work request W. Closes session.
- *NCM* – Calls cleaner for connection to device D for work request W
- *Custom code* – Calls out to ticketing system
- *Ticketing System* – Returns reason R for change
- *Custom code* – Calls NCM API to copy reason R into custom data on device

## Cleaner coding

NCM will use the configuration file appserver.rcx to determine what class to use for the session cleaner. A default do-nothing cleaner, `com.rendition.api.DefaultCleaner`, is provided by NCM.

To install your own cleaner, follow these steps:

- Code your own cleaner that implements the `com.rendition.api.Cleaner` interface
- modify "cleaner/className" option in appserver.rcx file, specifying your own class
- build a JAR file that contains all your new classes and copy it into %JBOSS_HOME%/server/default/lib directory

## Commands

This section provides information for issuing commands and receiving the correct result data types.

### Permissions

When invoked via the NCM Java API, the required user permissions for all commands are the same as for the Telnet/SSH Proxy interactive mode. The commands are documented in the “Permissions” section.

### Commands and Return Values

The following table lists the commands and return values.

| Command | Success Code | Return Value (s) | Asynchronous |
|---|---|---|---|
| activate device | 200 | null | |
| add advanced script | 200 | null | |
| add authentication | 200 | String | |
| add command script | 200 | null | |
| add device | 201 | null | |
| add device to group | 200 | null | |
| Add diagnostic | 200 | null | |
| add event | 200 | null | |
| add group | 200 | null | |
| Add group to parent group | 200 | null | |
| Add parent group | 200 | null | |
| add ip | 200 | null | |
| add system message | 200 | null | |
| add user | 207 | null | |
| annotate access | 200 | null | |
| annotate config | 200 | null | |
| configure syslog | 200 | null | |
| deactivate device | 200 | null | |
| del access | 200 | null | |
| del authentication | 200 | null | |
| del device | 200 | null | |
| del device data | 200 | null | |
| del device from group | 200 | null | |
| del drivers | 200 | null | |
| del event | 200 | null | |

| Command | Success Code | Return Value (s) | Asynchronous |
|---|---|---|---|
| del group | 200 | null | |
| Del group from parent group | 200 | null | |
| del ip | 200 | null | |
| del session | 200 | null | |
| del script | 200 | null | |
| del system message | 200 | null | |
| del task | 217 | null | |
| del user | 211 | null | |
| deploy config | 200 | null | √ |
| diff config | 200 | null | |
| discover driver | 200 | null | √ |
| discover drivers | 200 | null | √ |
| get snapshot | 200 | null | √ |
| list access | 200 | ResultSet | |
| list access all | 200 | ResultSet | |
| list basicip | 200 | Collection of String | |
| list config | 200 | ResultSet | |
| list config all | 200 | ResultSet | |
| list device | 501 | ResultSet | |
| list device data | 200 | ResultSet | |
| list deviceinfo | 200 | Collection of String | |
| list diagnostic | 200 | Collection of String | |
| list drivers | 200 | ResultSet | |
| list event | 200 | ResultSet | |
| list groups | 200 | ResultSet | |
| list icmp | 200 | Collection of String | |
| list int | 200 | Collection of String | |
| list ip | 200 | ResultSet | |
| list ip all | 200 | ResultSet | |
| list module | 200 | ResultSet | |
| list ospfneighbor | 200 | Collection of String | |
| list port | 200 | ResultSet | |
| list routing | 200 | Collection of String | |
| List script | 200 | ResultSet | |

| Command | Success Code | Return Value (s) | Asynchronous |
|---|---|---|---|
| list session | 200 | ResultSet | |
| list system message | 200 | ResultSet | |
| list task | 200 | ResultSet | |
| list task all | 513 | ResultSet | |
| list user | 511 | ResultSet | |
| Mod advanced script | 200 | String | |
| mod authentication | 200 | String | |
| Mod command script | 200 | String | |
| mod device | 204 | null | |
| Mod diagnostic | 200 | String | |
| mod group | 200 | null | |
| mod ip | 200 | String | |
| mod module | 200 | null | |
| mod port | 200 | null | |
| mod task | 215 | null | |
| mod unmanaged device | 200 | null | |
| mod user | 209 | null | |
| passwd | 200 | null | √ |
| pause polling | 200 | null | |
| ping | 200 | String | √ |
| reload server options | 200 | null | |
| resume polling | 200 | null | |
| run advanced script | 200 | null | |
| run command script | 200 | String | |
| run diagnostic | 200 | String | √ |
| run script | 200 | String | √ |
| show access | 200 | ResultSet | |
| show basicip | 200 | String | |
| show config | 200 | String | |
| show device | 200 | ResultSet | |
| show device config | 200 | String | |
| show device latest diff | 200 | String | |
| show deviceinfo | 200 | String | |
| show diagnostic | 200 | String | |

| Command | Success Code | Return Value (s) | Asynchronous |
|---|---|---|---|
| show event | 200 | ResultSet | |
| show fastlookup | 200 | String | |
| show group | 200 | ResultSet | |
| show icmp | 200 | String | |
| show int | 200 | String | |
| show ip | 200 | ResultSet | |
| show latest access | 200 | ResultSet | |
| show module | 200 | ResultSet | |
| show ospfneighbor | 200 | String | |
| show polling status | 200 | String | |
| show port | 200 | ResultSet | |
| show routing | 200 | String | |
| show script | 200 | String | |
| show session | 200 | ResultSet | |
| show session commands | 200 | String | |
| show snapshot | 200 | String | |
| show system message | 200 | ResultSet | |
| show task | 221 | ResultSet | |
| show user | 219 | ResultSet | |
| synchronize | 200 | String | √ |
| traceroute | 200 | String | √ |

## 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:

| Command | ResultSet Contents starting with column 1 |
|---|---|
| list device data<br>list config<br>list config all | java.lang.Integer deviceDataID<br>java.lang.String dataBlock<br>java.lang.String blockType<br>java.util.Date createDate<br>java.lang.String comments<br>java.lang.Integer deviceAccessLogID<br>java.lang.Short blockFormat |
| list drivers | java.lang.Integer driverLookupID<br>java.lang.Integer deviceID<br>java.lang.String baseModelName<br>java.lang.String driverName |

| Command | ResultSet Contents starting with column 1 |
|---|---|
| show access<br>list access<br>list access all<br>show latest access | java.lang.Integer deviceAccessLogID<br>java.lang.String displayName<br>java.lang.String actionTaken<br>java.lang.String accessTrigger<br>java.util.Date createDate<br>java.lang.Integer createUserID<br>java.lang.Integer interceptorLogID<br>java.lang.String comments<br>java.lang.Short noPrune<br>java.lang.String externalChangeRequestID<br>java.lang.Integer deviceID<br>java.lang.String changeEventData<br>java.lang.String deviceDataCustom1<br>java.lang.String deviceDataCustom2<br>java.lang.String deviceDataCustom3<br>java.lang.String deviceDataCustom4<br>java.lang.String deviceDataCustom5<br>java.lang.String deviceDataCustom6 |
| list device<br>show device | java.lang.Integer deviceID<br>java.lang.String primaryFQDN<br>java.lang.String hostName<br>java.lang.String primaryIPAddress<br>java.lang.String consoleIPAddress<br>java.lang.String nATIPAddress<br>java.lang.String tFTPServerIPAddress<br>java.lang.Integer consolePort<br>java.lang.String deviceName<br>java.lang.String serialNumber<br>java.lang.String assetTag<br>java.lang.String softwareVersion<br>java.lang.String firmwareVersion<br>java.lang.String vendor<br>java.lang.String model<br>java.lang.String deviceType<br>java.lang.String geographicalLocation<br>java.lang.String timeZone<br>java.lang.String deviceFunction<br>java.lang.String comments<br>java.util.Date createDate<br>java.util.Date lastAccessAttemptDate<br>java.util.Date lastAccessSuccessDate<br>java.util.Date lastSnapshotDate<br>java.lang.String lastAccessAttemptStatus<br>java.lang.Integer lastModifiedUserID<br>java.lang.Short excludeFromPoll<br>java.lang.Short canUseChangeAgents<br>java.lang.String accessMethods<br>java.lang.String modemNumber<br>java.lang.Short managementStatus |

| Command | ResultSet Contents starting with column 1 |
|---|---|
| | java.lang.String feedSource<br>java.util.Date lastImportDate<br>java.util.Date lastRecordModifiedDate<br>java.lang.String changeEventData<br>java.lang.Integer mostRecentConfigID<br>java.lang.Integer lastConfigChangeUserID<br>java.lang.Integer latestStartupRunningDiffer<br>java.lang.String deviceCustom1<br>java.lang.String deviceCustom2<br>java.lang.String deviceCustom3<br>java.lang.String deviceCustom4<br>java.lang.String deviceCustom5<br>java.lang.String deviceCustom6 |
| list groups<br>show group | java.lang.Integer deviceGroupID<br>java.lang.String deviceGroupName<br>java.util.Date createDate<br>java.lang.String comments<br>java.lang.String deviceGroupCustom1<br>java.lang.String deviceGroupCustom2<br>java.lang.String deviceGroupCustom3<br>java.lang.String deviceGroupCustom4<br>jjava.lang.String deviceGroupCustom5<br>java.lang.String deviceGroupCustom6<br>java.lang.Integer deviceCount |
| show session<br>list session | java.lang.Integer interceptorLogID<br>java.util.Date startDate<br>java.util.Date endDate<br>java.lang.Integer userID<br>java.lang.Integer deviceID<br>java.lang.String deviceIP<br>java.lang.String sessionType<br>java.lang.String sessionData<br>java.lang.Short status<br>java.lang.String interceptorLogCustom1<br>java.lang.String interceptorLogCustom2<br>java.lang.String interceptorLogCustom3<br>java.lang.String interceptorLogCustom4<br>java.lang.String interceptorLogCustom5<br>java.lang.String interceptorLogCustom6 |
| list system message<br>show system message | java.lang.Integer eventID<br>java.lang.Integer eventUserID<br>java.lang.Integer eventDeviceID<br>java.lang.String eventType<br>java.util.Date eventDate<br>java.lang.Short eventClass<br>java.lang.Integer eventTaskID<br>java.lang.String eventText |

| Command | ResultSet Contents starting with column 1 |
|---|---|
| list task<br>show task | java.lang.Integer scheduleTaskID<br>java.lang.Integer deviceGroupID<br>java.lang.Integer succeededChildCount<br>java.lang.Integer failedChildCount<br>java.lang.Integer pendingChildCount<br>java.lang.Integer parentTaskID<br>java.util.Date createDate<br>java.util.Date scheduleDate<br>java.lang.String comments<br>java.lang.Integer duration<br>java.lang.Short status<br>java.lang.String taskType<br>java.lang.Integer taskUserID<br>java.lang.Short retryCount<br>java.lang.Short retryInterval<br>java.lang.Short repeatType<br>java.lang.Short repeatWeekday<br>java.lang.Integer repeatInterval<br>java.lang.Integer deviceID<br>java.lang.Integer deviceDataID<br>java.lang.String result<br>java.lang.Short expensive<br>java.lang.String taskData<br>java.util.Date startDate<br>java.lang.Integer resultConfigID |
| list user<br>show user | java.lang.Integer userID<br>java.lang.String username<br>java.lang.String firstName<br>java.lang.String lastName<br>java.lang.String userPassword<br>java.lang.String emailAddress<br>java.util.Date createDate<br>java.lang.String timeZone<br>java.lang.Short requiredUser<br>java.lang.String aaaUserName<br>java.lang.String aaaPassword<br>java.lang.Short useAaaLoginForProxy<br>java.lang.String userCustom1<br>java.lang.String userCustom2<br>java.lang.String userCustom3<br>java.lang.String userCustom4<br>java.lang.String userCustom5<br>java.lang.String userCustom6 |
| show event<br>list event | java.lang.Integer eventID<br>java.lang.Integer eventUserID<br>java.lang.Integer eventDeviceID<br>java.lang.String eventType<br>java.util.Date eventDate |

| Command | ResultSet Contents starting with column 1 |
|---|---|
| | java.lang.Short eventClass<br>java.lang.Integer eventTaskID<br>java.lang.String eventText<br>java.lang.String eventData<br>java.lang.Integer configPolicyID |
| show ip<br>list ip<br>list ip all | java.lang.Integer ipID<br>java.lang.String ipValue<br>java.lang.String ipMask<br>java.lang.Integer ipPriority<br>java.lang.String ipName<br>java.lang.String comments<br>java.util.Date changeDate<br>java.lang.Short ipType<br>java.lang.Short usedToAccess<br>java.lang.Integer devicePortID<br>java.lang.Integer lastModifiedUserID<br>java.lang.Integer deviceID |
| show module<br>list module | Integer deviceModuleID<br>Integer deviceID<br>String slot<br>String moduleModel<br>String moduleDescription<br>String moduleOS<br>String firmwareVersion<br>String hardwareRevision<br>Integer memory<br>String moduleCustom1<br>String moduleCustom2<br>String moduleCustom3<br>String moduleCustom4<br>String moduleCustom5<br>String moduleCustom6<br>String comments<br>String serialNumber |
| show port<br>list port | Integer devicePortID<br>Integer deviceID<br>String portCustom1<br>String portCustom2<br>String portCustom3<br>String portCustom4<br>String comments<br>String portName<br>String portAllows<br>String portType<br>String portStatus<br>String description |

## Permissions

The following table describes user permissions that are required to execute the CLI commands described in the “Commands” section of this document. These roles are the default roles created by NCM. An administrator can create new permission groups and roles, and assign them to users.

**User Permissions Matrix**

| User | Description | Reconfigure Devices<br>Log into enable mode<br>View unmasked passwords<br>Run configuration scripts<br>Deploy configuration<br>Change passwords | System Administration<br>Highly Sensitive<br>Manage users<br>Delete historical information<br>Edit/delete any users's tasks<br>Define custom diagnostics | Other<br>Administrative settings<br>Authentication rules<br>View all telnet/SSH sessions | Group Tasks<br>Custom scripts & diagnostics<br>Snapshots & polling<br>Driver discovery<br>Syslog configuration<br>Password deployment<br>Import<br>FQDN lookup | Modify NCM Information<br>Devices<br>Groups<br>Configuration comments |
|---|---|---|---|---|---|---|
| **Admin** | Admins are highly trusted users responsible for administering the NCM application, managing users, setting policy, and running network-wide operations requiring a high degree of skill and care. They have permission to take any action in the NCM system on any device. | **All** Devices | X | X | X | X |
| **Power User** | Power users are highly trusted expert engineers allowed to perform most actions in the system. They can reconfigure and otherwise act on groups of devices in the system. They may be restricted as to which devices they have permission to reconfigure. | **Specified** Devices | | X | X | X |

**User Permissions Matrix**

| User | Description | Reconfigure Devices<br>Log into enable mode<br>View unmasked passwords<br>Run configuration scripts<br>Deploy configuration<br>Change passwords | System Administration<br>**Highly Sensitive**<br>Manage users<br>Delete historical information<br>Edit/delete any users's tasks<br>Define custom diagnostics | System Administration<br>**Other**<br>Administrative settings<br>Authentication rules<br>View all telnet/SSH sessions | **Group Tasks**<br>Custom scripts & diagnostics<br>Snapshots & polling<br>Driver discovery<br>Syslog configuration<br>Password deployment<br>Import<br>FQDN lookup | **Modify NCM Information**<br>Devices<br>Groups<br>Configuration comments |
|---|---|---|---|---|---|---|
| **Full Access** | Full Access users are qualified network engineers trusted with passwords to configure some or all devices in the network. They have permission to modify most information in the NCM database, and can reconfigure devices one-at-a-time but not in batch. They may be restricted as to which devices they have permission to reconfigure. | **Specified** Devices | | | | **X** |
| **Limited Access** | Limited Access users are operator users that do not have passwords to configure network devices. They have permission to view but not modify most information in NCM. Sensitive information such as device passwords will be masked out. They cannot run batch operations or operations which would reconfigure network devices. | **No** Devices | | | | |

# Appendix A: NCM Documentation

To open any of the available NCM documentation, on the on the menu bar click Docs. NCM also includes 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:

http://www.cisco.com/en/US/products/ps6923/tsd_products_support_series_ho me.html

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

To open any of the available documents, on the menu bar click Docs. The CiscoWorks Network Compliance Manager Documentation window 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.

- *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:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

# Appendix C: CLI/API Command Reference

---

## 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

add authentication -loc <Location> [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-snmpro <Read only community string(s)>] [-snmprw <Read write community string(s)>] [-snmpv3user <SNMPv3 Username>] [-snmpv3authpw <SNMPv3 Authentication Password>] [-snmpv3encryptpw <SNMPv3 Encryption Password>] [-user <Username>] [-passwd <Password>] [-enableuser <Enable username>] [-enablepasswd <Enable password>] [-connectionmethods <Connection methods>] [-accessvariables <Access variables>] [-start <Task start date>] [-appendsnmpro] [-appendsnmprw] [-sync] [-group <Group name>]

### 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-seperated 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-seperated 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 -snmpro 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

add device -ip <IP address> [-hostname <Host name>] [-comment <Comment>] [-description <Device name>] [-model <Device model>] [-vendor <Device vendor>] [-domain <Domain name>] [-serial <Serial number>] [-asset <Asset tag>] [-location <Location>] [-unmanaged <Unmanaged>] [-nopoll <Do not poll>] [-consoleip <Console IP address, if using console server>] [-consoleport <Console Port>] [-tftpserverip <TFTP server IP address, if using NAT>] [-natip <NAT IP address>] [-useconsoleserver <true or false>] [-accessmethods <Comma-separated list of access methods>] [-hierarchylayer <Hierarchy layer>]

### 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, snnp_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 supportes, 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 fooset -images c:\\data\\bar.bin -driver CiscoPIX
- add image -imageset fooset -images c:\\data\\bar.bin,c:\\images\\foobar.bin -model "WS-C2924M-XL-EN (C2900XL series)"

- add image -imageset fooset -images /var/upload/bar.bin,/var/upload/foo.bin -driver CiscoPIX

---

## add imageoption

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

### Synopsis

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

add ip -ipvalue <Value> [-deviceip <Device IP address>] [-comment <Comment>] [-usetoaccess <Use to Access Device>] [-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.
- -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 seperately 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 facilites.

**Synopsis**

configure syslog [-ip <IP address>] [-group <Groupname>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-rep <Task repeat period>] [-sync] [-start <Task start date>] [-comment <Snapshop comment>] [-usesyslogrelay <IP address>]

**Description**

Have the system configure the specified device to send all syslog messages necessary for the system's change detection facilites to function optimally to the system's syslog server. The configuration operation is atually 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 -host Zangief -start 2004:02:29:23:59 -rep 1weeks
- configure syslog -ip 192.0.2.10 -sync
- configure syslog -group mygroup

---

## connect

Connect to a device.

**Synopsis**

connect [-login] [-method <telnet|ssh|ssh1|ssh2|rlogin>] [-override] [-info] [-ignoreptyerrors] []

**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 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 data -cutoff 2004:02:29:23:59

---

## 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>]

**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 -metadataid 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 / Diagnositc ID>] [-name <Script / Diagnositc Name>] [-type <Script / Diagnositc 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

deploy config [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-id <Config ID>] [-configtext <Config Text>] [-start <Task start date>] [-sync] -option <Deployment option>

### 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**

Exit the the system.

**Examples**

- exit

## get snapshot

Get the config from a device.

### Synopsis

get snapshot [-ip <IP address>] [-group <Groupname>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-rep <Task repeat period>] [-sync] [-start <Task start date>] [-comment <Snapshot comment>] [-duration <Estimated duration of snapshot task.>]

### 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 -host Zangief -start 2004:02:29:23:59 -rep 2days
- 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 12:30:00YYYY-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 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 -deviecid 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 12:30:00YYYY-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.
- -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 12:30:00YYYY-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.
- -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

list device [-software <Software Version>] [-vendor <Device Vendor>] [-type <Device Type>] [-model <Device Model>] [-family <Device Family>] [-group <Device Group>] [-disabled] [-pollexcluded] [-ids <Device ID List>] [-hierarchy <Hierarchy Layer>] [-host <Device Host Name>] [-ip <Device IP Address>] [-realm <Realm Name>] [-startid <ID>] [-limitcount <Count>]

### 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**

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

**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**

list device id [-software <Software Version>] [-vendor <Device Vendor>] [-type <Device Type>] [-model <Device Model>] [-family <Device Family>] [-group <Device Group>] [-disabled] [-pollexcluded] [-id <Device ID>] [-host <Device Host Name>] [-ip <Device IP Address>] [-realm <Realm Name>] [-hierarchy <Hierarchy Layer>] [-viewable_by <Viewable By>]

**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 id -group "border routers" -disabled
- list device id -family "Cisco IOS"
- list device id -vendor Nortel
- list device id -viewable_by 201

---

## 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.

- -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.

- -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 12:30:00YYYY-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 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 RequiredApproval RequestApproval GrantedApproval Task ChangedApproval Task DeletedApproval DeniedApproval Task TimeoutApproval OverrideCommand Authorization ErrorUser Authentication ErrorConfiguration Policy AddedConfiguration Policy Non-ComplianceConfiguration Policy ChangedConfiguration Policy Pattern TimeoutConfiguration Rule AddedConfiguration Rule ChangedDevice Access FailureDevice AddedDevice Password ChangeDevice BootedDevice Command Script FailedDevice Command Script Completed SuccessfullyDevice Configuration ChangeDevice Configuration Change - No UserDevice Configuration Deployment FailureDevice Configuration DeploymentDevice Data FailureDevice DeletedDevice Diagnostic ChangedDevice Diagnostic FailedDevice Diagnostic Completed SuccessfullyDevice Flash Storage Running LowGroup ModifiedGroup AddedGroup DeletedDevice InaccessibleDevice EditedLast Used Device Password ChangedDevice ManagedDevice Missing from ImportDevice Permissions - ModifiedDevice Reservation ConflictDevice SnapshotDevice Software ChangeDevice Startup/Running Config DifferenceDevice UnmanagedSoftware Vulnerability DetectedEmail Report SavedExternal Directory Server Authentication ErrorLicense Almost ExceededLicense Almost ExpiredLicense ExceededLicense ExpiredModule AddedModule ChangedModule RemovedMonitor OkayMonitor ErrorDevice Permissions - New DeviceDevice Password Change FailureConcurrent Telnet/SSH Session OverrideReserved Device Configuration ChangedScheduled for Deploy Configuration EditedScheduled for Deploy Password ModifiedServer StartupSession Data CapturedSoftware Update FailedSoftware Update SucceededSummary Reports GeneratedPending Task DeletedTask StartedTicket CreatedUser LoginUser LogoutUser AddedUser DeletedUser Permission ChangedUser 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 12:30:00YYYY-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 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

- -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 12:30:00YYYY-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 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>]

**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 12:30:00YYYY-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 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 -deviceid 1401

---

### 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

list module [-model <Model Number>] [-type <Module Description>] [-firmware <Firmware Version>] [-hardware <Hardware Revision>] [-memory <Memory>] [-comment <Comment>] [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-group <Device Group Name>]

### 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 12:30:00YYYY-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 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 parition.

- -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 12:30:00YYYY-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 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.

- -host - A valid hostname
- -fqdn - A valid Fully Qualified Domain Name
- -deviceid - A device ID
- -id - A device ID
- -type - Type of the desired script or diagnostic - may be command, advanced or diagnostic

**Examples**

- list script mode
- list script mode -type diagnostic
- list script mode -ip 192.0.2.10
- list script mode -id 1420 -type advanced

---

## 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 12:30:00YYYY-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 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 12:30:00YYYY-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 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 -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>]

**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**

mod advanced script [-id <Script ID>] [-name <Script Name>] [-newname <New Name>] [-description <New Description>] [-scripttype <New Script Type>] [-family <New Device Family>] [-language <New Script Language>] [-parameters <New Parameters>] [-script <New Script Text>]

**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**

mod authentication -loc <Location> [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-snmpro <Read only community string(s)>] [-snmprw <Read write community string(s)>] [-snmpv3user <SNMPv3 Username>] [-snmpv3authpw <SNMPv3 Authentication Password>] [-snmpv3encryptpw <SNMPv3 Encryption Password>] [-user <Username>] [-passwd <Password>] [-enableuser <Enable username>] [-enablepasswd <Enable password>] [-connectionmethods <Connection methods>] [-accessvariables <Access variables>] [-start <Task start date>] [-appendsnmpro] [-appendsnmprw] [-sync] [-group <Group name>] [-rulename <Password Rule name>]

**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-seperated 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-seperated 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 authentication -loc device -ip 192.0.2.10 -passwd some -enablepasswd all -snmprw brillig,slithy,toves,gire -appendsnmprw -sync
- mod authentication -loc device -ip 192.0.2.10 -passwd less -enablepasswd more -snmpro foo,bar,fork,snork -start 2004:02:29:23:59
- mod authentication -loc group -group MyDevices -passwd less -enablepasswd more -snmpro foo,bar,fork,snork -start 2004:02:29:23:59

---

## mod command script

Modify an existing command script.

### Synopsis

mod command script [-id <Script ID>] [-name <Script Name>] [-newname <New Name>] [-description <New Description>] [-scripttype <New Script Type>] [-mode <New Mode>] [-driver <New Driver List>] [-script <New Script Text>]

### 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,CiscoIOSSwitch" -script "extended ping $Target_IP$"

---

## mod device

Modify the properties of a device.

### Synopsis

mod device [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-hostname <New Hostname>] [-comment <Comment>] [-description <Device name>] [-model <Device model>] [-vendor <Device vendor>] [-domain <Domain name>] [-serial <Serial number>] [-asset <Asset tag>] [-location <Location>] [-unmanaged <Unmanaged>] [-nopoll <Do not poll>] [-newIP <New IP address>] [-consoleip <Console IP address, if using console server>] [-consoleport <Console Port>] [-tftpserverip <TFTP server IP address, if using NAT>] [-natip <NAT IP address>] [-customname <Customname>] [-customvalue <Customvalue>] [-useconsoleserver <true or false>] [-accessmethods <Comma-separated list of access methods>] [-hierarchylayer <Hierarchy layer>]

### 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 REALM_NAME:, where REALM_NAME is the name of the Realm the address is in.
- -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**

mod diagnostic [-id <Diagnostic ID>] [-name <Diagnostic Name>] [-newname <New Name>] [-description <New Description>] [-mode <New Mode>] [-driver <New Driver List>] [-script <New Script Text>]

**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

mod ip -ipvalue <Value> [-deviceip <Device IP address>] [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-comment <Comment>] [-usetoaccess <Use to Access Device>]

### 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 coresponds 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 RN_DEVICE
- 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**

mod task -id <Task ID> [-comment <Comment>] [-retryInterval <Retry interval>] [-expensive] [-notexpensive] [-days <Days>] [-retryCount <Retry count>] [-repeatType <Repeat type>] [-duration <Duration>] [-start <Start>] [-repeatInterval <Repeat interval>] [-approve <Approval comment>] [-reject <Reason the task is not approved>] [-override <Reason for overriding approval process>] [-customname <Custom name>] [-customvalue <Custom value>]

**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, thur, 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 <Device ID> [-deviceportid <Device ID>] [-remotedeviceid <Device ID>] [-remotedeviceportid <Device ID>] [-serverid <Server ID>] [-serverportid <Server ID>]

**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

mod user -u <Username> [-p <Password>] [-fn <First name>] [-ln <Last name>] [-email <Email address>] [-priv <User Privilege>] [-newusername <Username>] [-aaausername <Username>] [-aaapassword <AAA Password>] [-useaaaloginforproxy <Use AAA Logins for Proxy (yes|no)>] [-extauthfailover <Allow External Auth Failover (yes|no)>] [-customname <Customname>] [-customvalue <Customvalue>] [-status <Enable or Disable the user (enable|disable)>]

### 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.
- -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).
- -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 sepecified 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 sepecified 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 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, or 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 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 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**

- 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

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**

run advanced script [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-group <Groupname>] -name <Script Name> [-parameters <Parameters>] [-variables <Variable List>] [-start <Task start date>] [-rep <Task repeat period>] [-sync] [-nowait] [-comment <Snapshop comment>] [-presnapshot <true or false>] [-postsnapshot <true, false or task>]

**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,varable2='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 "-l /usr/etc/log.txt" -sync

---

## run command script

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

**Synopsis**

run command script [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-group <Groupname>] -name <Script Name> [-variables <Variable List>] [-linebyline] [-start <Task start date>] [-rep <Task repeat period>] [-sync] [-nowait] [-comment <Snapshop comment>] [-presnapshot <true or false>] [-postsnapshot <true, false or task>]

**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,varable2='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'.

- -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 -start 2006:03:23:11:33 -startdir /usr/local/bin -resultfile /home/jdoe/out.log -app"echo foo"
- 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**

run script [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-group <Group Name>] [-mode <Command Script Mode>] -script <Command Script> [-rep <Task repeat period>] [-start <Task start date>] [-sync] [-nowait] [-comment <Run script comment>]

**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 congfiglet
- -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>]

**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 latest diff -ip 192.0.2.10
- show device latest diff -ip "East Site:192.0.2.10"

---

## 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.

- -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 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>]

### 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 / Diagnositc ID>] [-name <Script / Diagnositc Name>] [-type <Script / Diagnositc Type>]

**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 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 seperated 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 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**

synchronize [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-deviceid <Device ID>] [-group <Group Name>] [-skipinsync <Skip if Synchronized>] [-rep <Task repeat period>] [-start <Task start date>] [-sync] [-comment <Task comment>]

### 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 test 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

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

---