Java API Reference for Network Compliance Manager

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-10253-02

Java API Reference Guide for Network Compliance Manager

Table of Contents

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 1.2.1*, 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 1.2.1, 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 1.2.1. 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 list config list config all	java.lang.Integer deviceDataID java.lang.String dataBlock java.lang.String blockType java.util.Date createDate java.lang.String comments java.lang.Integer deviceAccessLogID java.lang.Short blockFormat
list drivers	java.lang.Integer driverLookupID java.lang.Integer deviceID java.lang.String baseModelName java.lang.String driverName

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

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

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

Command	ResultSet Contents starting with column 1
	java.lang.Short eventClass java.lang.Integer eventTaskID java.lang.String eventText java.lang.String eventData java.lang.Integer configPolicyID
show ip list ip list ip all	java.lang.Integer ipID java.lang.String ipValue java.lang.String ipMask java.lang.Integer ipPriority java.lang.String ipName java.lang.String comments java.util.Date changeDate java.lang.Short ipType java.lang.Short usedToAccess java.lang.Integer devicePortID java.lang.Integer lastModifiedUserID java.lang.Integer deviceID
show module list module	Integer deviceModuleID Integer deviceID String slot String moduleModel String moduleDescription String moduleOS String firmwareVersion String hardwareRevision Integer memory String moduleCustom1 String moduleCustom2 String moduleCustom3 String moduleCustom4 String moduleCustom5 String moduleCustom6 String comments String serialNumber
show port list port	Integer devicePortID Integer deviceID String portCustom1 String portCustom2 String portCustom3 String portCustom4 String comments String portName String portAllows String portType String portStatus 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	System Administration		Group Tasks	Modify NCM Information
			Highly Sensitive	Other		
		Log into enable mode View unmasked passwords Run configuration scripts Deploy configuration Change passwords	Manage users Delete historical information Edit/delete any users's tasks Define custom diagnostics	Administrative settings Authentication rules View all telnet/SSH sessions	Custom scripts & diagnostics Snapshots & polling Driver discovery Syslog configuration Password deployment Import FQDN lookup	Devices Groups 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 Log into enable mode View unmasked passwords Run configuration scripts Deploy configuration Change passwords	System Administration **Highly Sensitive** Manage users Delete historical information Edit/delete any users's tasks Define custom diagnostics	System Administration **Other** Administrative settings Authentication rules View all telnet/SSH sessions	**Group Tasks** Custom scripts & diagnostics Snapshots & polling Driver discovery Syslog configuration Password deployment Import FQDN lookup	**Modify NCM Information** Devices Groups 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.

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 1.2.1* — 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 1.2.1* — 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

---

activate device — Mark a device as activated.

Synopsis:

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

Description:

-ip — a.b.c.d where 0 <= a,b,c,d <= 255

- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name

Example:

activate device -ip 207.99.30.226

---

add advanced script —Add a new 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:

- - id <Script ID> — ID of the advanced script to edit.
- -name <Script Name> — Name of the advanced script to edit
- -newname <New Name> — New name for the script being modified.
- -description <New Description> — New description for the script being modified.
- -scripttype <New Script Type> — New script type (i.e. user defined subcategory).
- -family <New Device Family> — New device family for the script being modified.
- -language <New Script Language> — New language for the script being modified - must be a supported language such as Expect or Perl.
- -parameters <New Parameters> — New command line parameters for the script being modified.
- -script <New Script Text> — New script text.

Example:

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>] [-snmpro <Read only community string(s)>] [-snmprw <Read write community string(s)>] [-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 update what NCM knows of a device or network's password information. The -ip option provides information specific to the device. Otherwise, the command adds a network-wide password rule to NCM. 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 NCM's database. "device" tells the command that the password changes should be made on the device as well. "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: 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.
- snmpro — When used in conjunction with -loc db, this argument is taken as a single community string understood by NCM 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 NCM 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).
- 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 NCM to connect to devices. It 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.

Example:

add authentication -loc db -ip 207.99.30.226 -passwd fish -snmpro public -enablepasswd 31337

---

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> — Name for the new command script.
- -Description — The descriptive name of the new command script.
- Scripttype <Script Type> — Script type, for example a user-defined subcategory.
- -mode <Mode> — The command script mode.
- -driver <Driver List> —List of applicable drivers provided as a comma separated list of internal driver names.
- -script <Script Text> —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 $Target_IP$"

---

add device — Add a device to NCM.

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>]

Description:

- -ip — a.b.c.d where 0 <= a,b,c,d <= 255
- -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.
- -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 NCM. 1: Mark this device to be unmanaged by NCM.
- -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, SCP, FTP, TFTP, SNMP}. If this option is not provided, NCM tries all access methods when attempting to connect to the device.

Example:

add device -ip 207.99.30.226

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>] -group <Device group>

Description:

- -ip — a.b.c.d where 0 <= a,b,c,d <= 255
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name
- -group — The name of the device group to which the device should be added.

Example:

add device to group -ip 207.99.30.226 -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> — Name for the new diagnostic.
- -description <Description> — escription for the new diagnostic.
- -mode <Mode> — Command script mode
- -driver <Driver List> — List of applicable drivers - provided as a comma separated list of internal river names.
- -script <Script Text>— Dagnostic script text.

Example:

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

---

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 NCM 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.

Example:

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

---

add group — Add a group to NCM.

Synopsis:

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

Description:

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

Example:

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

Example:

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

---

add ip — Add new secondary ip.

Synopsis:

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

Description:

- -deviceip — The device's ip address a.b.c.d where 0 <= a,b,c,d <= 255
- -ipvalue — The ip value a.b.c.d where 0 <= a,b,c,d <= 255
- -comment — Additional information regarding the device.
- -usetoaccess — Use this ip Value to access its device, 0: yes, 1: no, default: no

Example:

add ip -deviceip 207.99.30.226 -ipvalue 207.99.23.23 -comment "my own ip"

---

add parent group — Add a parent group to NCM.

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.

Example:

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

---

add system message — Add a system message.

Synopsis:

add system message -message <System Message> [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>]

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
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name

Example:

add system message -ip 207.99.30.226 -message "Connectivity to the border router has been restored."

---

add user — Add a user to NCM.

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>]

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 (0=No,1=Yes).

Example:

add user -u johnd -p fish -fn john -ln doe -email johnd@nowhere.net.

---

annotate access — Modify the comments on, or the displayed 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.
- -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

Example:

annotate access -id 2 -comment "Device tainted at this point." -name "Intrusion detected"

---

annotate config — Add a comment to the specified config.

Synopsis:

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

Description:

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

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

Example:

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
- -id — A valid device id
- -name — A valid internal driver name, supported by system

Example:

assign driver -ip 207.99.30.226 -name CiscoIOSGenericNoLog

---

configure syslog — Configure a device to send syslog messages to NCM's change detection facilities.

Synopsis:

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

Description:

Have NCM configure the specified device to send all syslog messages necessary for NCM's change detection facilities to function optimally to NCM's syslog server.

- -ip — a.b.c.d where 0 <= a,b,c,d <= 255
- -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

- -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 want to setup forwarding on that relay host rather than have the device log directly to NCM syslog server. The specified IP address is taken to be the IP address of the relay host.

Example:

configure syslog -ip 207.99.30.226

---

connect — Connect to a device.

Synopsis:

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

Description:

Connect to a device through NCM' Proxy Interface via telnet or ssh. If you are connected to a device through a console server, you may hit ctrl-\ to return to the NCM 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 NCM or for devices in NCM 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.
- -Hostname, 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 NCM.

Example:

connect 207.99.30.226

---

deactivate device — Mark a device as deactivated.

Synopsis:

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

Description:

- -ip — a.b.c.d where 0 <= a,b,c,d <= 255
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name

Example:

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.

Example:

del access -id 6288

---

del authentication — Deletes all password information associated with the specified device.

Synopsis:

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

Description:

- -ip — a.b.c.d where 0 <= a,b,c,d <= 255: 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.

Example:

del authentication -ip 207.99.30.226

---

del device — Delete the specified device.

Synopsis:

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

Description:

- -ip — a.b.c.d where 0 <= a,b,c,d <= 255
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name

Example:

del device -ip 207.99.30.226

---

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.

Example:

del device data -id 866227436

---

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

Synopsis:

Deletes device from group [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] -group <Device group>

Description:

- -ip — a.b.c.d where 0 <= a,b,c,d <= 255
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name
- -group — The name of the device group from which the device should be deleted.

Example:

del device from group -ip 207.99.30.226 -group tech-dev

---

del device from parent group — Remove a device group from a parent device group.

Synopsis:

Deletes a device from parent group -parent <Parent group name> -child <Child group name>

Description:

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

Example:

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

---

del drivers — Delete all drivers associated with a device.

Synopsis:

deletes drivers [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>]

Description:

- -ip — a.b.c.d where 0 <= a,b,c,d <= 255
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name

Example:

del drivers -ip 207.99.30.226

---

del group — Delete a group from NCM.

Synopsis:

deletes 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.

Example:

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

---

del ip — Delete the specified ip.

Synopsis:

del ip -deviceip <Device IP address> -ipvalue <Value>

Description:

- -deviceip — The device's ip address a.b.c.d where 0 <= a,b,c,d <= 255
- -ipvalue — The ip value a.b.c.d where 0 <= a,b,c,d <= 255

Example:

del ip -deviceip 207.99.30.226 -ipvalue 207.99.31.23

---

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 <Script / Diagnositc ID> — ID of the desired script or diagnostic.
- -name <Script / Diagnositc Name> — Name of the desired script or diagnostic.
- -type <Script / Diagnositc 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

Example:

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

Example:

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

Example:

del task -id 4321

---

del user — Delete a user from NCM.

Synopsis:

del user -u <User name>

Description: -u -- The user name to be deleted

Example:

del user -u johnd

---

deploy config — Deploy the config to a device.

Synopsis:

deploy config [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] -id <Config ID> [-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
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name
- -id — The ID of the config 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.

Example:

deploy config -ip 207.99.30.226 -id 1962 -sync

---

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

Example:

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>]

Description:

- -ip — a.b.c.d where 0 <= a,b,c,d <= 255
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name

Example:

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>]

Description:

Attempts to match a driver to the specified device.

- -ip — a.b.c.d where 0 <= a,b,c,d <= 255: 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.

Example:

discover driver -ip 207.99.30.226

---

discover drivers — Discover drivers for all devices.

Synopsis:

discover drivers

Description:

Attempts to match a driver to each device that NCM recognizes.

Example:

discover drivers

---

enable device — Mark a device as enabled.

Synopsis:

enable device [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>]

Description:

- -ip — a.b.c.d where 0 <= a,b,c,d <= 255
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name

Example:

enable device -ip 207.99.30.226

---

exit — Exit NCM.

Synopsis:

exit

Description: Exit

Example:

exit

---

get snapshot — Get the config from a device.

Synopsis:

get snapshot [-ip <IP address>] [-group <Groupname>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-rep <Task repeat period>] [-sync] [-start <Task start date>] [-comment <Snapshop comment>]

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
- -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
- -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.

Example:

get snapshot -ip 207.99.30.226

---

import — Import device or device password information.

Synopsis:

import -input <Filename> -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 device password information contained in appropriately formatted CSV files. (Please contact Spport for CSV file format specifications.)

- -input — Contains CSV device or device password data.
- -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 NCM. Valid values are true | false
- -filter — An application that reads the input file from stdin, and writes a NCM 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 NCM' "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 NCM, but is not visible via any UI.

Example:

import -input devices.csv -data device -log import.log -append true -cleanafter false -deviceorigin "Border Routers" -filter prepro.exe

---

list access — List all access records for a device.

Synopsis:

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

Description:

- -ip — a.b.c.d where 0 <= a,b,c,d <= 255
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name
- -start — Display only those access records created on or after the given date. Values for this option can 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 2002/09/06YYYY:MM:DD:HH:MM (e.g. 2002:09:06:12:30), or, one of: now, today, yesterday, tomorrow; Or, in the format: e.g. 3 days ago is a positive integer is one of: seconds, minutes, hours, days, weeks, months, years;. is one of: ago, before, later, after.

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

Example:

list access -ip 207.99.30.226

---

list access all — List all access records for all devices.

Synopsis:

list access all

Description: list all

Example:

list access all

---

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>] [-start <Date>] [-end <Date>]

Description:

- -ip — a.b.c.d where 0 <= a,b,c,d <= 255
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name
- -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), or, one of: now, today, yesterday, tomorrow; Or, in the format: e.g. 3 days ago is a positive integer. is one of: seconds, minutes, hours, days, weeks, months, years;. is one of: ago, before, later, after.
- -end — Display only those configs stored on or before the given date. Values for this option have the same format as for the option -start.

Example:

list basicip -ip 207.99.30.226

---

list config — List all configs for the specified device.

Synopsis:

list config [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-start <Date>] [-end <Date>]

Description:

- -ip — a.b.c.d where 0 <= a,b,c,d <= 255
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name
- -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), or, one of: now, today, yesterday, tomorrow; Or, in the format: e.g. 3 days ago is a positive integer. is one of: seconds, minutes, hours, days, weeks, months, years;. is one of: ago, before, later, after.
- -end — Display only those configs stored on or before the given date. Values for this option have the same format as for the option -start.

Example:

list config -ip 207.99.30.226

---

list config all — List all configs for all devices.

Synopsis:

list config all

Description: list config all

Example:

list config all

---

list device — List devices.

Synopsis:

list device [-group <Device group>] [-disabled] [-pollexcluded]

Description:

Lists all devices in the system unless you include one of the options; with -group, the command lists all devices in the specified group, with -disabled lists unmanaged devices, with -pollexcluded list devices excluded from polling.

- -group — The name of the device group whose devices are to be listed.
- -disabled — List devices that are unmanaged.
- -pollexcluded — List devices excluded from polling.

Example:

list device

---

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
- -dataType — A string describing the type of device data record to list

Example:

list device data -ip 207.99.30.226

---

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>]

Description:

- -ip — a.b.c.d where 0 <= a,b,c,d <= 255
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name

Example:

list deviceinfo -ip 207.99.30.226

---

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>] [-start <Date>] [-end <Date>]

Description:

- -diagnostic — A diagnostic name
- -ip — a.b.c.d where 0 <= a,b,c,d <= 255
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name
- 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), or, one of: now, today, yesterday, tomorrow; Or, in the format: e.g. 3 days ago is a positive integer. is one of: seconds, minutes, hours, days, weeks, months, years;.is one of: ago, before, later, after.

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

Example:

list diagnostic -ip 207.99.30.226 -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>]

Description:

- -ip — a.b.c.d where 0 <= a,b,c,d <= 255
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name

Example:

list drivers -ip 207.99.30.226

---

list events — List all events.

Synopsis:

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

Description:

- -ip — a.b.c.d where 0 <= a,b,c,d <= 255 (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.)
- -type <type> — A valid event type. Refer to the *User Guide for Network Compliance Manager 1.2.1* for event descriptions.
- -start <Date> — List events after this date. Values for this option may be in one of the following formats:
  YYYY-MM-DD HH:MM:SS e.g. 2002-09-06 12:30:00
  YYYY-MM-DD HH:MM e.g. 2002-09-06 12:30
  YYYY-MM-DD e.g. 2002-09-06
  YYYY/MM/DD e.g. 2002/09/06
  YYYY:MM:DD:HH:MM e.g. 2002:09:06:12:30
  Or, one of: now, today, yesterday, tomorrow
  Or, in the format: <number> <time unit> <designator> e.g. 3 days ago
  <number> is a positive integer.

<time unit> is one of: seconds, minutes, hours, days, weeks, months, years;.
<designator> is one of: ago, before, later, after.

- -end <Date> — List events before this date.

Examples:

list event -ip 207.99.130.226

list event -start yesterday

list device -group "border routers"

---

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>] [-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
- -parent — List all device groups that are children of the indicated parent group

Example:

list groups -type device

---

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>] [-start <Date>] [-end <Date>]

Description:

- -ip — a.b.c.d where 0 <= a,b,c,d <= 255
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name
- -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; Or, one of: now, today, yesterday, tomorrow; Or, in the format: e.g. 3 days ago is a positive integer. is one of: seconds, minutes, hours, days, weeks, months, years;. is one of: ago, before, later, after.
- -end — Display only those ICMPTest models stored on or before the given date. Values for this option have the same format as for the option -start.

Example:

list icmp -ip 207.99.30.226

---

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>] [-start <Date>] [-end <Date>]

Description:

- -ip — a.b.c.d where 0 <= a,b,c,d <= 255
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name
- -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), or, one of: now, today, yesterday, tomorrow; Or, in the format: e.g. 3 days ago is a positive integer. is one of: seconds, minutes, hours, days, weeks, months, years;. is one of: ago, before, later, after.
- -end — Display only those ShowInterfaces models stored on or before the given date. Values for this option have the same format as for the option -start.

Example:

list int -ip 207.99.30.226

---

list ip — List ip.

Synopsis:

list ip -deviceip <Device IP address>

Description:

Lists ip addresses for specific device: -deviceip — The device's ip address a.b.c.d where 0 <= a,b,c,d <= 255

Example:

list ip -deviceip 207.99.30.226

---

list ip all — List all secondary ip.

Synopsis:

list ip all

Description: List all secondary ip addresses in the system.

Example:

list ip all

---

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>] [-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
- -group — List only device modules on all devices with this device group name

Example:

list module -host border7.red

---

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>] [-start <Date>] [-end <Date>]

Description:

- -ip — a.b.c.d where 0 <= a,b,c,d <= 255
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name
- -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; Or, one of: now, today, yesterday, tomorrow; Or, in the format: e.g. 3 days ago is a positive integer. is one of: seconds, minutes, hours, days, weeks, months, years;. is one of: ago, before, later, after.
- -end — Display only those ShowOSPFNeighbors models stored on or before the given date. Values for this option have the same format as for the option -start.

Example:

list ospfneighbor -ip 207.99.30.226

---

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>]

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

Example:

list port -host border7.red

---

list routing — List all routing tables for a device.

Synopsis:

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

Description:

- -ip — a.b.c.d where 0 <= a,b,c,d <= 255
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name
- -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; Or, one of: now, today, yesterday, tomorrow; Or, in the format: e.g. 3 days ago is a positive integer. is one of: seconds, minutes, hours, days, weeks, months, years;. is one of: ago, before, later, after.
- -end — Display only those routing tables stored on or before the given date. Values for this option have the same format as for the option -start.

Example:

list routing -ip 207.99.30.226

---

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

Synopsis:

list script [-type <Type>] [-scripttype <Script Type>]

Description:

- -type <Type> — Type of the desired script or diagnostic - may be command, advanced or diagnostic -scripttype <Script Type>
- User defined script type (i.e. subcategory) — applies only to command scripts and advanced scripts

Examples:

list script
list script -type diagnostic
list script -type advanced -scripttype "Core Provisioning Scripts"

---

list session — List all interceptor log records for a device.

Synopsis:

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

Description:

- -ip — a.b.c.d where 0 <= a,b,c,d <= 255
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name
- -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), or, one of: now, today, yesterday, tomorrow; Or, in the format: e.g. 3 days ago is a positive integer. is one of: seconds, minutes, hours, days, weeks, months, years;.is one of: ago, before, later, after.
- -end — Display only those interceptor log records created on or before the given date. Values for this option have the same format as for the option -start.

Example:

list session -ip 207.99.30.226

---

list system message — List system messages.

Synopsis:

list system message [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-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
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name
- -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; Or, one of: now, today, yesterday, tomorrow; Or, in the format: e.g. 3 days ago is a positive integer. is one of: seconds, minutes, hours, days, weeks, months, years;. is one of: ago, before, later, after.
- -end — Display only those system messages created on or before the given date. Values for this option have the same format as for the option -start.

Example:

list system message

---

list task — Display a list of tasks.

Synopsis:

list task [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-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 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: 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.
- -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 | completed): Display only those tasks with the specified status.
- -id — a task ID: Display the task with the given task ID.

Example:

list task -parentid 78

---

list task all — List all tasks.

Synopsis:

list task all

Description:

Equivalent to "list task".

Example:

list task all

---

list user — List all users.

Synopsis:

list user

Description:

Example:

list user

---

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 <Script ID> — ID of the advanced script to edit.
- -name <Script Name> — Name of the advanced script to edit.
- newname <New Name> — New name for the script being modified.
- description <New Description> — New description for the script being modified.
- scripttype <New Script Type> — New script type (i.e. user defined subcategory).
- family <New Device Family> — New device family for the script being modified.
- language <New Script Language — New language for the script being modified - must be a supported language (such as Expect or Perl).
- parameters <New Paramerters> — New command line parameters for the script being modified.
- script <New Script Text> — 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>] [-snmpro <Read only community string(s)>] [-snmprw <Read write community string(s)>] [-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 update what NCM knows of the device's password information. When using this command to modify passwords on a device or device group, the modification operation is 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 NCM' 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: 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.
- -snmpro — When used in conjunction with -loc db, this argument is taken as a single community string understood by NCM 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 NCM 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.)
- -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 NCM 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.

Example:

mod authentication -loc db -ip 207.99.30.226 -passwd fish -snmpro public -enablepasswd 31337

---

mod diagnostic — Modify an existing custom diagnostic script.

Synopsis:

mod diagnostic [-id <Diagnostic ID>] [-name <Diagnostic Name>] [-newname <NewName>] [-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, an error is reported and you must specify the unique diagnostic desired by ID.

- -id <Diagnostic ID> — ID of the diagnostic to edit.
- -name <Diagnostic Name> — Name of the diagnostic to edit.
- -newname <New Name> — New name for the diagnostic being modified.
- -description <New Description> — New description for the diagnostic being modified.
- -mode <New Mode> — New command script mode.
- -driver <New Driver List> — New list of applicable drivers - provided as a comma separated list of internal driver names.
- -script <New Script Text> — 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 10.1.34.115"

---

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 <Script ID> — ID of the command script to edit
- -name <Script Name> — Name of the command script to edit
- -newname <New Name> — New name for the script being modified
- -description <New Description — New description for the script being modified
- -scripttype <New Script Type> — New script type (i.e. user defined subcategory)
- -mode <New Mode> — New command script mode
- -driver <New Driver List> — New list of applicable drivers - provided as a comma separated list of internal driver names
- -script <New Script Text> — 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>] [-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>]

Description:

- -ip — a.b.c.d where 0 <= a,b,c,d <= 255
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name
- -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 NCM. 1: Mark this device to be unmanaged by NCM.
- -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; the new IP address of the device.
- -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
- -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, SCP, FTP, TFTP, SNMP}.

Example:

mod device -ip 207.99.30.226 -newIP 216.148.237.146

---

mod group — Modify a group.

Synopsis:

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

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

Example:

mod group -name "mystery routers" -type device -comment "removing these devices is a bad idea, but we don't really know what purpose they server."

---

mod ip — Modify the properties of a ip.

Synopsis:

mod ip -deviceip <Device IP address> -ipvalue <Value> [-comment <Comment>] [-usetoaccess <Use to Access Device>]

Description:

- -deviceip — The device's ip address a.b.c.d where 0 <= a,b,c,d <= 255
- -ipvalue — The ip value a.b.c.d where 0 <= a,b,c,d <= 255
- -comment — Additional information regarding the device.
- -usetoaccess — Use this IP Value to access its device, 0=yes, 1= no, default=no

Example:

mod ip -deviceip 207.99.30.226 -ipvalue 207.99.23.23 -comment "my own ip"

---

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

Example:

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>]

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 — How many seconds the task can run

- -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

Example:

mod task -id 7097 -repeatType 4 -days mon,wed,thur

---

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
- -comment — Additional information regarding the device.

Example:

mod unmanaged device -ip 207.99.30.226 -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>] [-customname <Customname>] [-customvalue <Customvalue>]

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 (0=No,1=Yes).
- -customname — The custom field name
- -customvalue — The custom field value

Example:

mod user -u johnd -p new -fn Johnathan -email jdoe@somewhere.nu

---

passwd — Change 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

Example:

passwd -oldpwd youroldpassword -newpwd yournewpwd

---

pause polling — Stop polling.

Synopsis:

pause polling

Description:

Example:

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 exectued 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.

Example:

ping -source 207.99.30.226 -dest 209.67.27.248

---

quit — Exit NCM.

Synopsis:

quit

Description:

Example:

quit

---

reload server options — Reload server options.

Synopsis:

reload server options

Description:

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

Example:

reload server options

---

resume polling — Resume polling.

Synopsis:

resume polling

Description:

Example:

resume polling

---

run advanced script — Run an exsiting advanced script on a device or group of devices.

Synopsis:

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

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 NCM task.

- -ip — a.b.c.d where 0 <= a,b,c,d <= 255
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name
- -group — A name of a device group (mutually exclusive with -ip, -host, or -fqdn)
- -mode — A command script mode to run the script in.
- -variables <Variable List> — 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. (For example: "variable1=value1,varable2='this is ''value 2'''.)
- -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 <Snapshot comment> — An optional comment about the snapshot.

Examples:

run advanced script -ip 207.99.30.226 -name "Extended Ping" –variables "Target_IP=10.121.53.7" -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'" -linebyline -sync

---

run diagnostic — Run a diagnostic on a device.

Synopsis:

run diagnostic [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-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
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name
- -group — A name of a device group (mutually exclusive with -ip, -host, or -fqdn)
- -diagnostic — A diagnostic to run. Built-in diagnostics are 'ONA Routing Table', 'ONA Interfaces' and 'ONA 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.

Example:

run diagnostic -ip 207.99.30.226 -diagnostic "vlan report" -sync

---

run command script — Run an command script on a device or group of devices.

Synopsis:

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

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 NCM task.

- -ip — a.b.c.d where 0 <= a,b,c,d <= 255
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name
- -group — A name of a device group (mutually exclusive with -ip, -host, or -fqdn)
- -mode — A command script mode to run the script in.
- -variables <Variable List> — 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. (For example: "variable1=value1,varable2='this is ''value 2'''.)
- -linebyline — Indicates that line-by-line deployment is preferred, rather than file-based deployment.
- -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 script being run.

Examples:

run command script -ip 207.99.30.226 -name "Extended Ping" –variables "Target_IP=10.121.53.7" -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 script — Run a command script on a device.

Synopsis:

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

- -ip — a.b.c.d where 0 <= a,b,c,d <= 255
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name
- -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.
- -comment — An optional comment about the script being run.

Example:

run script -ip 207.99.30.226 -mode "Cisco IOS enable" -script "show ver" -sync

---

show access — Display a device access record.

Synopsis:

show access -id <Device access record ID>

Description:

- -id — Specifies a device access record.

Example:

show access -id 510

---

show basicip — Show a BasicIP model.

Synopsis:

show basicip [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-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
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name
- -id — A config ID

Example:

show basicip -ip 207.99.30.226

---

show config — Show the contents of a config.

Synopsis:

show config -id <Config ID>

Description:

- -id — The ID of a config

Example:

show config -id 2600

---

show device — Show a device's properties.

Synopsis:

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

Description:

- -ip — a.b.c.d where 0 <= a,b,c,d <= 255
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name
- -id — A device ID

Example:

show device -ip 207.99.30.226

---

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>]

Description:

- -ip — a.b.c.d where 0 <= a,b,c,d <= 255
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name

Example:

show device config -ip 207.99.30.226

---

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>]

Description:

- -ip — a.b.c.d where 0 <= a,b,c,d <= 255
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name

Example:

show device latest diff -ip 207.99.30.226

---

show deviceinfo — Show a DeviceInformation model.

Synopsis:

show deviceinfo [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-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 Device Information 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
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name
- -id — A config ID

Example:

show deviceinfo -ip 207.99.30.226

---

show diagnostic — Show a diagnostic's results.

Synopsis:

show diagnostic -id <Config ID>

Description:

- -id — A config ID

Example:

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>]

Description:

- -ip — a.b.c.d where 0 <= a,b,c,d <= 255
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name

Example:

show driver -ip 207.99.30.226

---

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.

Example:

show group -name johnd

---

show icmp — Show an ICMPTest model.

Synopsis:

show icmp [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-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
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name
- -id — A config ID

Example:

show icmp -ip 207.99.30.226

---

show int — Show a ShowInterfaces model.

Synopsis:

show int [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-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
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name
- -id — A config ID

Example:

show int -ip 207.99.30.226

---

show ip — Show a ip's properties.

Synopsis:

show ip -deviceip <Device IP address> -ipvalue <Value>

Description:

- -deviceip — The device's ip address a.b.c.d where 0 <= a,b,c,d <= 255
- -ipvalue — The ip value a.b.c.d where 0 <= a,b,c,d <= 255

Example:

show ip -deviceip 207.99.30.226 -ipvalue 207.99.23.23

---

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>]

Description:

- -ip — a.b.c.d where 0 <= a,b,c,d <= 255
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name

Example:

show latest access -ip 207.99.30.226

---

show module — Show a module's properties.

Synopsis:

show module -id <Module ID>

Description:

- -id — The ID of a module

Example:

show module -id 527

---

show ospfneighbor — Show a ShowOSPFNeighbors model.

Synopsis:

show ospfneighbor [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-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
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name
- -id — A config ID

Example:

show ospfneighbor -ip 207.99.30.226

---

show polling status — Show the current status of polling.

Synopsis:

show polling status

Example:

show polling status

---

show port — Show a port's properties.

Synopsis:

show port -id <Port ID>

Description:

- -id — The ID of a port

Example:

show port -id 527

---

show routing — Display a routing table .

Synopsis:

show routing [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-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
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name
- -id — A routing table ID

Example:

show routing -host rtr6.vfm.lab

Show Script — Show one command script, advanced script, or diagnostic.

Synopis:

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, an error will be reported. You must specify the unique script by ID.

- -id <Script / Diagnositc ID> — ID of the desired script or diagnostic.
- -name <Script / Diagnositc Name> — Name of the desired script or diagnostic.
- -type <Script / Diagnositc Type> — Type of the desired script or diagnostic.

Examples:

show script -id 5

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

---

show session — Show interceptor log record.

Synopsis:

show session -id <Interceptor log id>

Description:

- -id — Interceptor log ID

Example:

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

Example:

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>]

Description:

- -ip — a.b.c.d where 0 <= a,b,c,d <= 255
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name

Example:

show snapshot -ip 207.99.30.226

---

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

Example:

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

Example:

show task -id 354

---

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

Example:

show user -u johnd

---

ssh — Make an ssh connection to a device .

Synopsis:

ssh [-override]

Description:

Connect to a device through NCM' 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 NCM 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, 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 NCM.

Example:

ssh 207.99.30.226

---

synchronize — Synchronize a device's startup and running configs.

Synopsis:

synchronize [-ip <IP address>] [-host <Hostname>] [-fqdn <Fully Qualified Domain Name>] [-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
- -host — A valid hostname
- -fqdn — A valid Fully Qualified Domain Name
- -group — A name of a device group (mutually exclusive with -ip, -host, or -fqdn)
- -skipinsync — Indicates that the command should skip any device that NCM 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.

Example:

synchronize -ip 207.99.30.226 -sync

---

telnet — Make a telnet connection to a device.

Synopsis:

telnet [-override]

Description:

Connect to a device through NCM' 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 NCM shell after logging out of the device.

- -overrid Force a connection to a device in the event that simultaneous connection warning or prevention is turned on.
- -Hostname, 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 NCM.

Example:

telnet 207.99.30.226

---

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 exectued 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, 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 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.

Example:

traceroute -source 207.99.30.226 -dest 209.67.27.248

---

version — List NCM version.

Synopsis:

version

Description:

Example:

version