



Cisco Networking Services Configuration Guide, Cisco IOS XE Everest 16.6

First Published: 2008-07-11

Last Modified: 2013-03-11

Americas 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: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/go/trademarks>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2008-2013 Cisco Systems, Inc. All rights reserved.



CONTENTS

CHAPTER 1

[Read Me First](#) 1

CHAPTER 2

[Configuring Cisco Networking Services](#) 3

[Finding Feature Information](#) 3

[Prerequisites for Cisco Networking Services](#) 3

[Restrictions for Cisco Networking Services](#) 4

[Information About Cisco Networking Services](#) 5

[Cisco Networking Services](#) 5

[Cisco Networking Services EXEC Agent](#) 5

[Cisco Networking Services Results Messages](#) 5

[Cisco Networking Services Message Formats](#) 6

[Cisco Networking Services IDs](#) 9

[Cisco Networking Services Password](#) 9

[Cisco Networking Services Zero Touch](#) 9

[How to Configure Cisco Networking Services](#) 10

[Deploying the Cisco Networking Services Device](#) 10

[Configuring Advanced Cisco Networking Services Features](#) 13

[Troubleshooting Cisco Networking Services Agents](#) 14

[Configuration Examples for Cisco Networking Services](#) 18

[Example: Deploying the Cisco Networking Services Device](#) 18

[Example: Using the Cisco Networking Services Zero Touch Solution](#) 18

[Additional References](#) 21

[Feature Information for Cisco Networking Services](#) 22

CHAPTER 3

[CNS Configuration Agent](#) 23

[Finding Feature Information](#) 23

[Information About CNS Configuration Agent](#) 23

[Cisco Networking Services Configuration Agent](#) 23

- Initial Cisco Networking Services Configuration 23
- Incremental Cisco Networking Services Configuration 24
- Synchronized Configuration 24
- How to Configure CNS Configuration Agent 25
 - Configuring the Cisco Networking Services Event and EXEC Agents 25
- Configuration Examples for CNS Configuration Agent 27
 - Example: Enabling and Configuring Cisco Networking Services Agents 27
 - Example: Retrieving a Cisco Networking Services Image from a Server 28
- Additional References 28
- Feature Information for CNS Configuration Agent 29

CHAPTER 4

- Cisco Networking Services Config Retrieve Enhancement with Retry and Interval 31**
 - Finding Feature Information 31
 - Information About CNS Config Retrieve Enhancement with Retry and Interval 31
 - Cisco Networking Services Config Retrieve Enhancement with Retry and Interval 31
 - How to Configure CNS Config Retrieve Enhancement with Retry and Interval 32
 - Retrieving a Cisco Networking Services Configuration from a Server 32
 - Configuration Examples for CNS Config Retrieve Enhancement with Retry and Interval 33
 - Example: Retrieving a Cisco Networking Services Configuration from a Server 33
 - Additional References 34
 - Feature Information for CNS Config Retrieve Enhancement with Retry and Interval 35

CHAPTER 5

- Cisco Networking Services Interactive CLI 37**
 - Finding Feature Information 37
 - Information About CNS Interactive CLI 37
 - Cisco Networking Services Interactive CLI 37
 - Additional References 38
 - Feature Information for CNS Interactive CLI 38

CHAPTER 6

- Command Scheduler (Kron) 41**
 - Finding Feature Information 41
 - Restrictions for Command Scheduler 41
 - Information About Command Scheduler (Kron) 42
 - Command Scheduler 42
 - How to Configure Command Scheduler (Kron) 42

Configuring Command Scheduler Policy Lists and Occurrences	42
Troubleshooting Tips	45
Configuration Examples for Command Scheduler (Kron)	45
Example: Command Scheduler Policy Lists and Occurrences	45
Additional References	46
Feature Information for Command Scheduler (Kron)	47

CHAPTER 7

Network Configuration Protocol	49
Finding Feature Information	49
Prerequisites for NETCONF	49
Information About NETCONF	50
NETCONF Notifications	50
How to Configure NETCONF	50
Configuring the NETCONF Network Manager Application	50
Delivering NETCONF Payloads	51
Formatting NETCONF Notifications	53
Monitoring and Maintaining NETCONF Sessions	56
Configuration Examples for NETCONF	57
Example: Configuring the NETCONF Network Manager Application	57
Example: Monitoring NETCONF Sessions	58
Additional References for NETCONF	60
Feature Information for NETCONF	61
Glossary	62

CHAPTER 8

NETCONF over SSHv2	65
Finding Feature Information	65
Prerequisites for NETCONF over SSHv2	65
Restrictions for NETCONF over SSH	66
Information About NETCONF over SSHv2	66
NETCONF over SSHv2	66
How to Configure NETCONF over SSHv2	67
Enabling SSH Version 2 Using a Hostname and Domain Name	67
Enabling SSH Version 2 Using RSA Key Pairs	68
Starting an Encrypted Session with a Remote Device	70
Troubleshooting Tips	70

What to Do Next	71
Verifying the Status of the Secure Shell Connection	71
Enabling NETCONF over SSHv2	72
Configuration Examples for NETCONF over SSHv2	73
Example: Enabling SSHv2 Using a Hostname and Domain Name	73
Enabling Secure Shell Version 2 Using RSA Keys Example	74
Starting an Encrypted Session with a Remote Device Example	74
Configuring NETCONF over SSHv2 Example	74
Additional References for NETCONF over SSHv2	75
Feature Information for NETCONF over SSHv2	77

CHAPTER 9

NETCONF Access for Configurations over BEEP	79
Finding Feature Information	79
Prerequisites for NETCONF Access for Configurations over BEEP	80
Restrictions for NETCONF Access for Configurations over BEEP	80
Information About NETCONF Access for Configurations over BEEP	80
NETCONF over BEEP Overview	80
How to Configure NETCONF Access for Configurations over BEEP	81
Configuring an SASL Profile	81
Enabling NETCONF over BEEP	83
Configuration Examples for NETCONF Access for Configurations over BEEP	86
Example: Enabling NETCONF over BEEP	86
Additional References for NETCONF Access for Configurations over BEEP	86
Feature Information for NETCONF Access for Configurations over BEEP	87



Read Me First

Important Information about Cisco IOS XE 16

Effective Cisco IOS XE Release 3.7.0E (for Catalyst Switching) and Cisco IOS XE Release 3.17S (for Access and Edge Routing) the two releases evolve (merge) into a single version of converged release—the Cisco IOS XE 16—providing one release covering the extensive range of access and edge products in the Switching and Routing portfolio.

Feature Information

Use [Cisco Feature Navigator](#) to find information about feature support, platform support, and Cisco software image support. An account on Cisco.com is not required.

Related References

- [Cisco IOS Command References, All Releases](#)

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see [What's New in Cisco Product Documentation](#).

To receive new and revised Cisco technical content directly to your desktop, you can subscribe to the [What's New in Cisco Product Documentation RSS feed](#). RSS feeds are a free service.



CHAPTER 2

Configuring Cisco Networking Services

The Cisco Networking Services (CNS) feature is a collection of services that can provide remote event-driven configuring of Cisco IOS networking devices and remote execution of some command-line interface (CLI) commands.

- [Finding Feature Information, page 3](#)
- [Prerequisites for Cisco Networking Services, page 3](#)
- [Restrictions for Cisco Networking Services, page 4](#)
- [Information About Cisco Networking Services, page 5](#)
- [How to Configure Cisco Networking Services, page 10](#)
- [Configuration Examples for Cisco Networking Services, page 18](#)
- [Additional References, page 21](#)
- [Feature Information for Cisco Networking Services, page 22](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Cisco Networking Services

- Configure the remote device to support the Cisco Networking Services configuration agent and the Cisco Networking Services event agent.

- Configure a transport protocol on the remote device that is compatible with the remote device's external interface. The following table lists the supported transport protocols that can be used depending on the device interface.
- Create the configuration template in the Cisco Networking Services configuration-engine provisioning database. (This task is best done by a senior network designer.)

Table 1: Device Interface and Transport Protocols Required by Cisco Networking Services Services

Device Interface	SLARP Transport Protocol	ATM InARP Transport Protocol	PPP (IPCP) Transport Protocol
T1	Yes	Yes	Yes
ADSL	No	Yes	Yes
Serial	Yes	No	Yes

Restrictions for Cisco Networking Services

Cisco Networking Services Configuration Engine

- The Cisco Networking Services configuration engine must be the Cisco Intelligence Engine 2100 (Cisco IE2100) series and must be running software version 1.3.
- The configuration engine must have access to an information database of attributes for building a configuration. This database can reside on the Cisco IE2100 itself.
- Configuration templates must be prepared on the Cisco Networking Services configuration engine before installation of the remote device.
- The user of Cisco Networking Services Flow-Through Provisioning and the Cisco Networking Services configuration engine must be familiar with designing network topologies, designing configuration templates, and using the Cisco Networking Services configuration engine.

Remote Device

- The remote device must run a Cisco IOS image that supports the Cisco Networking Services configuration agent and Cisco Networking Services event agent.
- Ports must be prepared on the remote device for connection to the network.
- You must ensure that the remote device is configured using Cisco Configuration Express.

Information About Cisco Networking Services

Cisco Networking Services

Cisco Networking Services is a foundation technology for linking users to networking services and provides the infrastructure for the automated configuration of large numbers of network devices. Many IP networks are complex with many devices, and each device must currently be configured individually. When standard configurations do not exist or have been modified, the time involved in initial installation and subsequent upgrading is considerable. The volume of smaller, more standardized, customer networks is also growing faster than the number of available network engineers. Internet service providers (ISPs) now need a method for sending out partial configurations to introduce new services. To address all these issues, Cisco Networking Services has been designed to provide “plug-and-play” network services using a central directory service and distributed agents. Cisco Networking Services features include Cisco Networking Services configuration and event agents and a Flow-Through Provisioning structure. The configuration and event agents use a Cisco Networking Services configuration engine to provide methods for automating initial Cisco device configurations, incremental configurations, and synchronized configuration updates, and the configuration engine reports the status of the configuration load as an event to which a network monitoring or workflow application can subscribe. The Cisco Networking Services Flow-Through Provisioning uses the Cisco Networking Services configuration and event agents to provide an automated workflow, eliminating the need for an on-site technician.

Cisco Networking Services EXEC Agent

The CNS EXEC agent allows a remote application to execute an EXEC mode CLI command on a Cisco device by sending an event message that contains the command. A restricted set of EXEC **show** commands is supported.

Cisco Networking Services Results Messages

When a partial configuration has been received by the device, each line of the configuration will be applied in the same order as it was received. If the Cisco parser has an error with one of the lines of the configuration, then all the configuration up to this point will be applied to the device, but none of the configuration beyond the error will be applied. If an error occurs, the **cns config partial** command will retry until the configuration successfully completes. In the pull mode, the command will not retry after an error. By default, NVRAM will be updated except when the **no-persist** keyword is configured.

A message will be published on the Cisco Networking Services event bus after the partial configuration is complete. The Cisco Networking Services event bus will display one of the following status messages:

- **cisco.mgmt.cns.config.complete**—Cisco Networking Services configuration agent successfully applied the partial configuration.
- **cisco.mgmt.cns.config.warning**—Cisco Networking Services configuration agent fully applied the partial configuration, but encountered possible semantic errors.
- **cisco.mgmt.cns.config.failure (CLI syntax)**—Cisco Networking Services configuration agent encountered a command line interface (CLI) syntax error and was not able to apply the partial configuration.

- `cisco.mgmt.cns.config.failure` (CLI semantic)—Cisco Networking Services configuration agent encountered a CLI semantic error and was not able to apply the partial configuration.

With the CNS Enhanced Results Messages feature, a second message is sent to the subject “`cisco.cns.config.results`” in addition to the appropriate message above. The second message contains both overall and line-by-line information about the configuration that was sent and the result of the action requested in the original message. If the action requested was to apply the configuration, then the information in the results message is semantic in nature. If the action requested was to check syntax only, then the information in the results message is syntactical in nature.

Cisco Networking Services Message Formats

SOAP Message Format

Using the Service-Oriented Access Protocol (SOAP) protocol provides a way to format the layout of Cisco Networking Services messages in a consistent manner. SOAP is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment. SOAP uses extensible markup language (XML) technologies to define an extensible messaging framework that provides a message format that can be exchanged over a variety of underlying protocols.

Within the SOAP message structure, there is a security header that enables Cisco Networking Services notification messages to authenticate user credentials.

Cisco Networking Services messages are classified into three message types: request, response and notification. The formats of these three message types are defined below.

Request Message

The following is the format of a Cisco Networking Services request message to the Cisco device:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope xmlns:SOAP="http://www.w3.org/2003/05/soap-envelope">
  <SOAP:Header>
    <wsse:Security xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext"
SOAP:mustUnderstand="0">
      <wsse:usernameToken>
        <wsse:Username>john</wsse:Username>
        <wsse:Password>cisco</wsse:Password>
      </wsse:usernameToken>
    </wsse:Security>
    <cns:cnsHeader version="1.0" xmlns:cns="http://www.cisco.com/management/cns/envelope">
      <cns:Agent>CNS_CONFIG</cns:Agent>
      <cns:Request>
        <cns:correlationID>IDENTIFIER</cns:correlationID>
        <cns:ReplyTo>
          <cns:URL>http://10.1.36.9:80/cns/ResToServer</cns:URL>
        </cns:ReplyTo>
      </cns:Request>
      <cns:Time>2003-04-23T20:27:19.847Z</cns:Time>
    </cns:cnsHeader>
  </SOAP:Header>
  <SOAP:Body xmlns="http://www.cisco.com/management/cns/config">
    <config-event config-action="read" no-syntax-check="TRUE">
      <config-data>
        <config-id>AAA</config-id>
        <cli>access-list 1 permit any</cli>
      </config-data>
    </config-event>
  </SOAP:Body>
</SOAP:Envelope>
```

**Note**

The ReplyTo field is optional. In the absence of the ReplyTo field, the response to the request will be sent to the destination where the request originated. The body portion of this message contains the payload and is processed by the Cisco Networking Services agent mentioned in the Agent field.

Response Message

The following is the format of a Cisco Networking Services response message from the Cisco device as a response to a request:

```
?xml version="1.0" encoding="UTF-8"?
SOAP:Envelope xmlns:SOAP="http://www.w3.org/2003/05/soap-envelope"
SOAP:Header
wsse:Security xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext"
SOAP:mustUnderstand="true"
wsse:UsernameToken
wsse:Username infysj-7204-8 /wsse:Username
wsse:Password NTM3NTg2NzIzOTg2MTk2MjgzNQ==/wsse:Password
/wsse:UsernameToken /wsse:Security
CNS:cnsHeader Version="2.0" xmlns:CNS="http://www.cisco.com/management/cns/envelope"
CNS:Agent CNS_CONFIG /CNS:Agent
CNS:Response
CNS:correlationID IDENTIFIER /CNS:correlationID
/CNS:Response
CNS:Time 2005-06-23T16:27:36.185Z /CNS:Time
/CNS:cnsHeader
/SOAP:Header
SOAP:Body xmlns="http://www.cisco.com/management/cns/config"
config-success config-id AAA /config-id /config-success
/SOAP:Body
/SOAP:Envelope
```

**Note**

The value of CorrelationId is echoed from the corresponding request message.

The body portion of this message contains the response from the Cisco device to a request. If the request is successfully processed, the body portion contains the value of the response put in by the agent that processed the request. If the request cannot be successfully processed, then the body portion will contain an error response.

Notification Message

The following is the format of a Cisco Networking Services notification message sent from the Cisco device:

```
?xml version="1.0" encoding="UTF-8"?
SOAP:Envelope xmlns:SOAP="http://www.w3.org/2003/05/soap-envelope"
SOAP:Header
wsse:Security xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext"
SOAP:mustUnderstand="true"
wsse:UsernameToken
wsse:Username dvlpr-7200-2 /wsse:Username
wsse:Password /wsse:Password
/wsse:UsernameToken
/wsse:Security
CNS:cnsHeader version="2.0" xmlns:CNS="http://www.cisco.com/management/cns/envelope"
CNS:Agent CNS_CONFIG_CHANGE/CNS:Agent
CNS:Notify /CNS:Notify
CNS:Time 2006-01-09T18:57:08.441Z/CNS:Time
/CNS:cnsHeader
/SOAP:Header
SOAP:Body xmlns="http://www.cisco.com/management/cns/config-change"
configChanged version="1.1" sessionData="complete"
sequence lastReset="2005-12-11T20:18:39.673Z" 7 /sequence
```

```

changeInfo
user/user
async port con_0 /port /async
when
absoluteTime 2006-01-09T18:57:07.973Z /absoluteTime
/when
/changeInfo
changeData
changeItem
context /context
enteredCommand
cli access-list 2 permit any /cli
/enteredCommand
oldConfigState
cli access-list 1 permit any /cli
/oldConfigState
newConfigState
cli access-list 1 permit any /cli
cli access-list 2 permit any /cli
/newConfigState
/changeItem
/changeData
/configChanged
/SOAP:Body
/SOAP:Envelope

```

A notification message is sent from the Cisco device without a corresponding request message when a configuration change is made. The body of the message contains the payload of the notification and it may also contain error information. If the request message sent to the Cisco device fails in XML parsing and the CorrelationId field cannot be parsed, then an error notification message will be sent instead of an error response.

Error Reporting

Error is reported in the body of the response or a notification message in the SOAP Fault element. The following is the format for reporting errors.

```

?xml version="1.0" encoding="UTF-8"?
SOAP:Envelope xmlns:SOAP="http://www.w3.org/2003/05/soap-envelope"
SOAP:Header
wsse:Security xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext"
SOAP:mustUnderstand="true"
wsse:UsernameToken
wsse:Username dvlpr-7200-2 /wsse:Username
wsse:Password /wsse:Password
/wsse:UsernameToken
/wsse:Security
CNS:cnsHeader version="2.0" xmlns:CNS="http://www.cisco.com/management/cns/envelope"
CNS:Agent CNS_CONFIG /CNS:Agent
CNS:Response
CNS:correlationID SOAP_IDENTIFIER /CNS:correlationID
/CNS:Response
CNS:Time 2006-01-09T19:10:10.009Z /CNS:Time
/CNS:cnsHeader
/SOAP:Header
SOAP:Body xmlns="http://www.cisco.com/management/cns/config"
SOAP:Detail
config-failure
config-id AAA /config-id
error-info
line-number 1 /line-number
error-message CNS_INVALID_CLI_CMD /error-message
/error-info
/config-failure
/SOAP:Detail
/SOAP:Fault
/SOAP:Body
/SOAP:Envelope

```

Cisco Networking Services IDs

The Cisco Networking Services ID is a text string that is used exclusively with a particular Cisco Networking Services agent. The Cisco Networking Services ID is used by the Cisco Networking Services agent to identify itself to the server application with which it communicates. For example, the Cisco Networking Services configuration agent will include the configuration ID when communicating between the networking device and the configuration server. The configuration server uses the Cisco Networking Services configuration ID as a key to locate the attribute containing the Cisco CLI configuration intended for the device that originated the configuration pull.

The network administrator must ensure a match between the Cisco Networking Services agent ID as defined on the routing device and the Cisco Networking Services agent ID contained in the directory attribute that corresponds to the configuration intended for the routing device. Within the routing device, the default value of the Cisco Networking Services agent ID is always set to the hostname. If the hostname changes, the Cisco Networking Services agent ID also changes. If the Cisco Networking Services agent ID is set using the CLI, any change will be followed by a message sent to syslog or an event message will be sent.

The Cisco Networking Services agent ID does not address security issues.

Cisco Networking Services Password

The Cisco Networking Services password is used to authenticate the Cisco Networking Services device. You must configure the Cisco Networking Services password the first time a device is deployed, and the Cisco Networking Services password must be the same as the bootstrap password set on the Configuration Engine (CE). If both the device and the CE bootstrap password use their default settings, a newly deployed device will be able to connect to the CE. Once connected, the CE manages the Cisco Networking Services password. Network administrators must ensure not to change the Cisco Networking Services password. If the Cisco Networking Services password is changed, connectivity to the CE will be lost.

Cisco Networking Services Zero Touch

The Cisco Networking Services Zero Touch feature provides a zero touch deployment solution where the device contacts a Cisco Networking Services configuration engine to retrieve its full configuration automatically. This capability is made possible through a single generic bootstrap configuration file common across all service provider end customers subscribing to the services. Within the Cisco Networking Services framework, customers can create this generic bootstrap configuration without device-specific or network-specific information such as interface type, line type, or controller type (if applicable).

The Cisco Networking Services connect functionality is configured with a set of Cisco Networking Services connect templates. A Cisco Networking Services connect profile is created for connecting to the Cisco Networking Services configuration engine and to implement the Cisco Networking Services connect templates on a Customer Premise Equipment (CPE) device. Cisco Networking Services connect variables can be used as placeholders within a Cisco Networking Services connect template configuration. These variables, such as the active DLCI, are substituted with real values before the Cisco Networking Services connect templates are sent to the device's parser.

To use the zero touch functionality, the device that is to be initialized must have a generic bootstrap configuration. This configuration includes Cisco Networking Services connect templates, Cisco Networking Services connect profiles, and the **ens config initial** command. This command initiates the Cisco Networking Services connect function.

The Cisco Networking Services connect functionality performs multiple ping iterations through the device's interfaces and lines, as well as any available controllers. For each iteration, the Cisco Networking Services connect function attempts to ping the Cisco Networking Services configuration engine. If the ping is successful, the pertinent configuration information can be downloaded from the Cisco Networking Services configuration engine. If connectivity to the Cisco Networking Services configuration engine is unsuccessful, the Cisco Networking Services connect function removes the configuration applied to the selected interface, and the Cisco Networking Services connect process restarts with the next available interface specified by the Cisco Networking Services connect profile.

The Cisco Networking Services Zero Touch feature provides the following benefits:

- Ensures consistent Cisco Networking Services commands.
- Use of a channel service unit (E1 or T1 controller) is allowed.

How to Configure Cisco Networking Services

Deploying the Cisco Networking Services Device

Incremental or partial configuration allows the remote device to be incrementally configured after its initial configuration. You must perform these configurations manually through the Cisco Networking Services configuration engine. The registrar allows you to change the configuration templates, edit parameters, and submit the new configuration to the device without a software or hardware restart.

Before You Begin

Perform this task to manually install an initial Cisco Networking Services configuration.

Your remote device arrives from the factory with a bootstrap configuration. Upon initial power-on, the device automatically pulls a full initial configuration from the Cisco Networking Services configuration engine, although you can optionally arrange for this manually as well. After initial configuration, you can optionally arrange for periodic incremental (partial) configurations for synchronization purposes.

For more details on using the Cisco CNS configuration engine to automatically install the initial CNS configuration, see the *Cisco CNS Configuration Engine Administrator's Guide* at http://www.cisco.com/en/US/docs/net_mgmt/configuration_engine/1.3/administration/guide/ag13.html

Initial Cisco Networking Services Configuration

Initial configuration of the remote device occurs automatically when the device is initialized on the network. Optionally, you can perform this configuration manually.

Cisco Networking Services assigns the remote device a unique IP address or hostname. After resolving the IP address (using Serial Line Address Resolution Protocol (SLARP), ATM Inverse ARP (ATM InARP), or PPP protocols), the system optionally uses Domain Name System (DNS) reverse lookup to assign a hostname to the device and invokes the Cisco Networking Services agent to download the initial configuration from the Cisco Networking Services configuration engine.

Incremental Configuration

Before you can configure an incremental configuration, Cisco Networking Services must be operational and the required Cisco Networking Services agents configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cns template connect** *name*
4. **cli** *config-text*
5. Repeat Step 4 to add all required CLI commands.
6. **exit**
7. **cns connect** *name* [**retry-interval** *interval-seconds*] [**retries** *number-retries*] [**timeout** *timeout-seconds*] [**sleep** *sleep-seconds*]
8. Do one of the following:
 - **discover** {**line** *line-type* | **controller** *controller-type* | **interface** [*interface-type*]}
 - **template** *name*
9. **exit**
10. **cns config initial** {*host-name* | *ip-address*} [**encrypt**] [*port-number*] [**page** *page*] [**syntax-check**] [**no-persist**] [**source** *interface name*] [**status** *url*] [**event**] [**inventory**]
11. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	cns template connect <i>name</i> Example: Device(config)# cns template connect template 1	Enters Cisco Networking Services template connect configuration mode and defines the name of a Cisco Networking Services connect template.
Step 4	cli <i>config-text</i> Example: Device(config-templ-conn)# cli encapsulation ppp	Specifies commands to configure the interface.

	Command or Action	Purpose
Step 5	Repeat Step 4 to add all required CLI commands. Example: Device(config-templ-conn)# cli ip directed-broadcast	Repeat Step 4 to add other CLI commands to configure the interface or to configure the modem lines.
Step 6	exit Example: Device(config-templ-conn)# exit	Exits Cisco Networking Services template connect configuration mode and completes the configuration of a Cisco Networking Services connect template. Note Entering the exit command is required. This requirement was implemented to prevent accidentally entering a command without the cli command.
Step 7	cns connect <i>name</i> [retry-interval <i>interval-seconds</i>] [retries number-retries] [timeout <i>timeout-seconds</i>] [sleep <i>sleep-seconds</i>] Example: Device(config)# cns connect profile-1 retry-interval 15 timeout 90	Enters Cisco Networking Services connect configuration mode and defines the parameters of a Cisco Networking Services connect profile for connecting to the Cisco Networking Services configuration engine.
Step 8	Do one of the following: <ul style="list-style-type: none"> • discover {line <i>line-type</i> controller <i>controller-type</i> interface [<i>interface-type</i>]} • template <i>name</i> Example: Device(config-cns-conn)# discover interface serial Example: Device(config-cns-conn)# template template-1	(Optional) Configures a generic bootstrap configuration. <ul style="list-style-type: none"> • discover —Defines the interface parameters within a Cisco Networking Services connect profile for connecting to the Cisco Networking Services configuration engine. or <ul style="list-style-type: none"> • template —Specifies a list of Cisco Networking Services connect templates within a Cisco Networking Services connect profile to be applied to a device's configuration.
Step 9	exit Example: Device(config-cns-conn)# exit	Exits Cisco Networking Services connect configuration mode and returns to global configuration mode.
Step 10	cns config initial { <i>host-name</i> <i>ip-address</i> } [encrypt] [<i>port-number</i>] [page <i>page</i>] [syntax-check] [no-persist] [source <i>interface name</i>] [status <i>url</i>] [event] [inventory]	Starts the Cisco Networking Services configuration agent, connects to the Cisco Networking Services configuration engine, and initiates an initial configuration. You can use this command only before the system boots for the first time.

	Command or Action	Purpose
	<p>Example:</p> <pre>Device(config)# cns config initial 10.1.1.1 no-persist</pre>	<p>Note The optional encrypt keyword is available only in images that support Secure Socket Layer (SSL).</p> <p>Caution If you write the new configuration to NVRAM by omitting the no-persist keyword, the original bootstrap configuration is overwritten.</p>
Step 11	<p>exit</p> <p>Example:</p> <pre>Device(config)# exit</pre>	Exits global configuration mode and returns to privileged EXEC mode.

Configuring Advanced Cisco Networking Services Features

Perform this task to configure more advanced Cisco Networking Services features. After the Cisco Networking Services agents are operational, you can configure some other features. You can enable the Cisco Networking Services inventory agent--that is, send an inventory of the device's line cards and modules to the Cisco Networking Services configuration engine--and enter Cisco Networking Services inventory mode.

Some other advanced features allow you to use the Software Developer's Toolkit (SDK) to specify how Cisco Networking Services notifications should be sent or how to access MIB information. Two encapsulation methods can be used: either nongranular (SNMP) encapsulation or granular (XML) encapsulation.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cns mib-access encapsulation {snmp | xml[size bytes]}**
4. **cns notifications encapsulation {snmp | xml}**
5. **cns inventory**
6. **transport event**
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>enable</p> <p>Example:</p> <pre>Device> enable</pre>	<p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	cns mib-access encapsulation {snmp xml[size bytes]} Example: Device(config)# cns mib-access encapsulation snmp	(Optional) Specifies the type of encapsulation to use when accessing MIB information. <ul style="list-style-type: none"> • Use the snmp keyword to specify that nongranular encapsulation is used to access MIB information. • Use the xml keyword to specify that granular encapsulation is used to access MIB information. The optional size keyword specifies the maximum size for response events, in bytes. The default byte value is 3072.
Step 4	cns notifications encapsulation {snmp xml} Example: Device(config)# cns notifications encapsulation xml	(Optional) Specifies the type of encapsulation to use when sending Cisco Networking Services notifications. <ul style="list-style-type: none"> • Use the snmp keyword to specify that nongranular encapsulation is used when Cisco Networking Services notifications are sent. • Use the xml keyword to specify that granular encapsulation is used when Cisco Networking Services notifications are sent.
Step 5	cns inventory Example: Device(config)# cns inventory	Enables the Cisco Networking Services inventory agent and enters Cisco Networking Services inventory mode. <ul style="list-style-type: none"> • An inventory of the device's line cards and modules is sent to the Cisco Networking Services configuration engine.
Step 6	transport event Example: Device(cns-inv)# transport event	Specifies that inventory requests are sent out with each Cisco Networking Services inventory agent message.
Step 7	exit Example: Device(cns-inv)# exit	Exits Cisco Networking Services inventory mode and returns to global configuration mode. <ul style="list-style-type: none"> • Repeat this command to return to privileged EXEC mode.

Troubleshooting Cisco Networking Services Agents

This section explains how to troubleshoot Cisco Networking Services agent issues.

The **show** commands created for the Cisco Networking Services image agent display information that is reset to zero after a successful reload of the device. Depending on the configuration of the image distribution process, the new image may not reload immediately. When a reload is not immediate or has failed, use the Cisco Networking Services image agent **show** commands to determine whether the image agent has connected to the image distribution server over HTTP or whether the image agent is receiving events from an application over the Cisco Networking Services Event Bus.

SUMMARY STEPS

1. **enable**
2. **show cns image status**
3. **clear cns image status**
4. **show cns image connections**
5. **show cns image inventory**
6. **debug cns image [agent| all| connection| error]**
7. **show cns event connections**
8. **show cns event subject [name]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables higher privilege levels, such as privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	show cns image status Example: Device# show cns image status	(Optional) Displays information about the Cisco Networking Services image agent status.
Step 3	clear cns image status Example: Device# clear cns image status	(Optional) Clears Cisco Networking Services image agent status statistics.
Step 4	show cns image connections Example: Device# show cns image connections	(Optional) Displays information about Cisco Networking Services image management server HTTP or HTTPS connections.
Step 5	show cns image inventory Example: Device# show cns image inventory	(Optional) Displays inventory information about the Cisco Networking Services image agent. <ul style="list-style-type: none"> • This command displays a dump of XML that would be sent out in response to an image agent inventory request message. The

	Command or Action	Purpose
		XML output can be used to determine the information requested by an application.
Step 6	debug cns image [agent all connection error] Example: Device# debug cns image all	(Optional) Displays debugging messages for Cisco Networking Services image agent services.
Step 7	show cns event connections Example: Device# show cns event connections	(Optional) Displays the status of the Cisco Networking Services event agent connection--such as whether it is connecting to the gateway, connected, or active--and to display the gateway used by the event agent and its IP address and port number.
Step 8	show cns event subject [name] Example: Device# show cns event subject subject1	(Optional) Displays a list of subjects of the Cisco Networking Services event agent that are subscribed to by applications.

Examples

In the following example, status information about the Cisco Networking Services image agent is displayed using the **show cns image status** privileged EXEC command:

```
Device# show cns image status
Last upgrade started at 11:45:02.000 UTC Mon May 6 2003
Last upgrade ended at 11:56:04.000 UTC Mon May 6 2003 status SUCCESS
Last successful upgrade ended at 11:56:04.000 UTC Mon May 6 2003
Last failed upgrade ended at 06:32:15.000 UTC Wed Apr 16 2003
Number of failed upgrades: 2
Number of successful upgrades: 6
  messages received: 12
  receive errors: 5
Transmit Status
  TX Attempts:4
  Successes:3      Failures 2
```

In the following example, information about the status of the Cisco Networking Services image management HTTP connections is displayed using the **show cns image connections** privileged EXEC command:

```
show cns image connections
CNS Image Agent: HTTP connections
Connection attempts 1
never connected:0  Abrupt disconnect:0
Last successful connection at 11:45:02.000 UTC Mon May 6 2003
```

In the following example, information about the Cisco Networking Services image agent inventory is displayed using the **show cns image inventory** privileged EXEC command:

```
show cns image inventory
Inventory Report
imageInventoryReport deviceName imageID Router /imageID hostName Router /ho
```

```
IOS (tm) C2600 Software (C2600-I-M), Experimental Version 12.3(20030414:081500)
Copyright (c) 1986-2003 by cisco Systems, Inc.
Compiled Mon 14-Apr-03 02:03 by engineer /versionString imageFile tftp://10.25.2.1.
```

In the following example, debugging messages for all Cisco Networking Services image agent services are displayed using the **debug cns imageprivileged EXEC** command. The Cisco Networking Services image agent in this example is connecting to an image server over HTTP. After connecting, the image server asks for an inventory of the Cisco device.

```
Device# debug cns image all
All cns image debug flags are on
Device# cns image retrieve

May  7 06:11:42.175: CNS Image Agent: set EXEC lock
May  7 06:11:42.175: CNS Image Agent: received message from EXEC
May  7 06:11:42.175: CNS Image Agent: set session lock 1
May  7 06:11:42.175: CNS Image Agent: attempting to send to
destination(http://10.1.36.8:8080/imgsrv/xgate):
?xml version="1.0" encoding="UTF-8"? cnsMessageversion="1.0" senderCredentials userName
dvlpr-7200-6 /userName /senderCredentials
messageID dvlpr-7200-6_2 /messageID sessionControl imageSessionStart version="1.0"
initiatorInfotrigger EXEC/trigger initiatorCredentials userName dvlpr-7200-6/userName
/initiatorCredentials /initiatorInfo /imageSessionStart /sessionControl /cnsMessage
May  7 06:11:42.175: CNS Image Agent: clear EXEC lock
May  7 06:11:42.175: CNS Image Agent: HTTP message sent url:http://10.1.36.8:8080/imgsrv/xgate
May  7 06:11:42.191: CNS Image Agent: response data alloc 4096 bytes
May  7 06:11:42.191: CNS Image Agent: HTTP req data free
May  7 06:11:42.191: CNS Image Agent: response data freed
May  7 06:11:42.191: CNS Image Agent: receive message
?xml version="1.0" encoding="UTF-8"?
cnsMessage version="1.0"
senderCredentials
userName myImageServer.cisco.com/userName
password R0lGODlhcgGSALMAAAQCAEMmCZtuMFQxDS8b/password
/senderCredentials
messageID dvlpr-c2600-2-476456/messageID
request
replyTo
serverReply http://10.1.36.8:8080/imgsrv/xgate /serverReply
/replyTo
imageInventory
inventoryItemList
all/
/inventoryItemList
/imageInventory
/request
/cnsMessage
```

The following example displays the IP address and port number of the primary and backup gateways:

```
Device# show cns event connections
The currently configured primary event gateway:
  hostname is 10.1.1.1.
  port number is 11011.
Event-Id is Internal test1
Keepalive setting:
  none.
Connection status:
  Connection Established.
The currently configured backup event gateway:
  none.
The currently connected event gateway:
  hostname is 10.1.1.1.
  port number is 11011.
```

The following sample displays a list of subjects of the Cisco Networking Services event agent that are subscribed to by applications:

```
Device# show cns event subject
The list of subjects subscribed by applications.
  cisco.cns.mibaccess:request
```

```
cisco.cns.config.load
cisco.cns.config.reboot
cisco.cns.exec.cmd
```

Configuration Examples for Cisco Networking Services

Example: Deploying the Cisco Networking Services Device

The following example shows an initial configuration on a remote device. The hostname of the remote device is the unique ID. The Cisco Networking Services configuration engine IP address is 172.28.129.22.

```
cns template connect templatel
cli ip address negotiated
cli encapsulation ppp
cli ip directed-broadcast
cli no keepalive
cli no shutdown
exit
cns connect host1 retry-interval 30 retries 3
exit
hostname RemoteRouter
ip route 172.28.129.22 255.255.255.0 10.11.11.1
cns id Ethernet 0 ipaddress
cns config initial 10.1.1.1 no-persist
exit
```

Example: Using the Cisco Networking Services Zero Touch Solution

Configuring PPP on a Serial Interface

The following example shows the bootstrap configuration for configuring PPP on a serial interface:

```
cns template connect ppp-serial
cli ip address negotiated
cli encapsulation ppp
cli ip directed-broadcast
cli no keepalive
exit
cns template connect ip-route
cli ip route 10.0.0.0 0.0.0.0 ${next-hop}
exit
cns connect serial-ppp ping-interval 1 retries 1
discover interface serial
template ppp-serial
template ip-route
exit
hostname 26ML
cns config initial 10.1.1.1 no-persist inventory
```

Configuring PPP on an Asynchronous Interface

The following example shows the bootstrap configuration for configuring PPP on an asynchronous interface:

```
cns template connect async
cli modem InOut
.
.
.
```



```

exit
cns template connect async-interface
cli encapsulation ppp
cli ip unnumbered FastEthernet0/0
cli dialer rotary-group 0
exit
cns template connect ip-route
cli ip route 10.0.0.0 0.0.0.0 ${next-hop}
exit
cns connect async
discover line Async
template async
discover interface
template async-interface
template ip-route
exit
hostname async-example
cns config initial 10.1.1.1 no-persist inventory

```

Configuring HDLC on a Serial Interface

The following example shows the bootstrap configuration for configuring High-Level Data Link Control (HDLC) on a serial interface:

```

cns template connect hdlc-serial
cli ip address slarp retry 1
exit
cns template connect ip-route
cli ip route 0.0.0.0 0.0.0.0 ${next-hop}
exit
cns connect hdlc-serial ping-interval 1 retries 1
discover interface serial
template hdlc-serial
template ip-route
exit
hostname host1
cns config initial 10.1.1.1 no-persist inventory

```

Configuring Aggregator Device Interfaces

The following examples show how to configure a standard serial interface and a serial interface bound to a controller on an aggregator device (also known as the DCE). In order for connectivity to be established, the aggregator device must have a point-to-point subinterface configured.

Standard Serial Interface

```

interface Serial0/1
  no ip address
  encapsulation frame-relay
  frame-relay intf-type dce
exit
interface Serial0/1.1 point-to-point
  10.0.0.0 255.255.255.0
  frame-relay interface-dlci 8

```

Serial Interface Bound to a Controller

```

controller T1 0
  framing sf
  linecode ami
  channel-group 0 timeslots 1-24
exit
interface Serial0:0
  no ip address
  encapsulation frame-relay

```

```

    frame-relay intf-type dce
  exit
  interface Serial0:0.1 point-to-point
    ip address ip-address mask
    frame-relay interface-dlci dlci

```

Configuring IP over Frame Relay

The following example shows the bootstrap configuration for configuring IP over Frame Relay on a CPE device:

```

cns template connect setup-frame
  cli encapsulation frame-relay
  exit
cns template connect ip-over-frame
  cli frame-relay interface-dlci ${dlci}
  cli ip address dynamic
  exit
cns template connect ip-route
  cli ip route 10.0.0.0 0.0.0.0 ${next-hop}
  exit
cns connect ip-over-frame
  discover interface Serial
  template setup-frame
  discover dlci
  template ip-over-frame
  template ip-route
  exit
cns config initial 10.1.1.1

```

Configuring IP over Frame Relay over T1

The following example shows the bootstrap configuration for configuring IP over Frame Relay over T1 on a CPE device:

```

cns template connect setup-frame
  cli encapsulation frame-relay
  exit
cns template connect ip-over-frame
  cli frame-relay interface-dlci ${dlci}
  cli ip address dynamic
  exit
cns template connect ip-route
  cli ip route 0.0.0.0 0.0.0.0 ${next-hop}
  exit
cns template connect t1-controller
  cli framing esf
  cli linecode b8zs
  cli channel-group 0 timeslots 1-24 speed 56
  exit
cns connect ip-over-frame-over-t1
  discover controller T1
  template t1-controller
  discover interface
  template setup-frame
  discover dlci
  template ip-over-frame
  template ip-route
  exit
cns config initial 10.1.1.1

```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco Networking Services commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples.	Cisco IOS Cisco Networking Services Command Reference
Cisco Networking Services Configuration Engine	Cisco CNS Configuration Engine Administrator Guide, 1.3

Standards and RFCs

Standard/RFC	Title
No new or modified standards/RFCs are supported by this feature, and support for existing standards/RFCs has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Cisco Networking Services

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 2: Feature Information for Cisco Networking Services

Feature Name	Releases	Feature Information
Cisco Networking Services	Cisco IOS XE Release 2.1 12.2(25)S 12.2(33) SRA 12.2(33)SB 12.2(33)SXI	<p>The Cisco Networking Services feature is a collection of services that can provide remote event-driven configuring of Cisco IOS networking devices and remote execution of some CLI commands.</p> <p>The following commands were introduced or modified by this feature: clear cns config stats, clear cns counters, clear cns event stats, cli (cns), cns config cancel, cns config initial, cns config notify, cns config partial, cns config retrieve, cns connect, cns event, cns exec, cns id, cns template connect, cns trusted-server, debug cns config, debug cns exec, debug cns xml-parser, logging cns-events, show cns config stats, show cns event connections, show cns event stats, show cns event subject.</p>



CNS Configuration Agent

- [Finding Feature Information, page 23](#)
- [Information About CNS Configuration Agent, page 23](#)
- [How to Configure CNS Configuration Agent, page 25](#)
- [Configuration Examples for CNS Configuration Agent, page 27](#)
- [Additional References, page 28](#)
- [Feature Information for CNS Configuration Agent, page 29](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About CNS Configuration Agent

Cisco Networking Services Configuration Agent

The Cisco Networking Services configuration agent is involved in the initial configuration and subsequent partial configurations on a Cisco device. To activate the Cisco Networking Services configuration agent, enter any of the **cns config** CLI commands.

Initial Cisco Networking Services Configuration

When a routing device first comes up, it connects to the configuration server component of the Cisco Networking Services configuration agent by establishing a TCP connection through the use of the **cns config**

initial command, a standard CLI command. The device issues a request and identifies itself by providing a unique configuration ID to the configuration server.

When the Cisco Networking Services web server receives a request for a configuration file, it invokes the Java servlet and executes the corresponding embedded code. The embedded code directs the Cisco Networking Services web server to access the directory server and file system to read the configuration reference for this device (configuration ID) and template. The Configuration Agent prepares an instantiated configuration file by substituting all the parameter values specified in the template with valid values for this device. The configuration server forwards the configuration file to the Cisco Networking Services web server for transmission to the routing device.

The Cisco Networking Services configuration agent accepts the configuration file from the Cisco Networking Services web server, performs XML parsing, checks syntax (optional), and loads the configuration file. The routing device reports the status of the configuration load as an event to which a network monitoring or workflow application can subscribe.

For more details on using the Cisco Cisco Networking Services configuration engine to automatically install the initial Cisco Networking Services configuration, see the *Cisco Networking Services Configuration Engine Administrator's Guide* at <http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/cns/cc/rel13/ag13/index.htm>.

Incremental Cisco Networking Services Configuration

Once the network is up and running, new services can be added using the Cisco Networking Services configuration agent. Incremental (partial) configurations can be sent to routing devices. The actual configuration can be sent as an event payload by way of the event gateway (push operation) or as a signal event that triggers the device to initiate a pull operation.

The routing device can check the syntax of the configuration before applying it. If the syntax is correct, the routing device applies the incremental configuration and publishes an event that signals success to the configuration server. If the device fails to apply the incremental configuration, it publishes an event that indicates an error.

Once the routing device has applied the incremental configuration, it can write the configuration to NVRAM or wait until signaled to do so.

Synchronized Configuration

When a routing device receives a configuration, the device has the option to defer application of the configuration upon receipt of a write-signal event. The Cisco Networking Services Configuration Agent feature allows the device configuration to be synchronized with other dependent network activities.

How to Configure CNS Configuration Agent

Configuring the Cisco Networking Services Event and EXEC Agents

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cns config partial** {*host-name* | *ip-address*} [**encrypt**] [*port-number*] [*source interface name*] [**inventory**]
4. **logging cns-events** [*severity-level*]
5. **cns exec** [**encrypt**] [*port-number*] [**source** {*ip-address* | *interface-type-number*}]
6. **cns event** {*hostname* | *ip-address*} [**encrypt**] [*port-number*] [**backup**] [**failover-time** *seconds*] [**keepalive** *seconds* *retry-count*] [**source** *ip-address* | *interface-name*][**clock-timeout** *time*] [**reconnect-time** *time*]
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	cns config partial { <i>host-name</i> <i>ip-address</i> } [encrypt] [<i>port-number</i>] [<i>source interface name</i>] [inventory] Example: Device(config)# cns config partial 172.28.129.22 80	(Optional) Starts the Cisco Networking Services configuration agent, which provides Cisco Networking Services configuration services to Cisco clients, and initiates an incremental (partial) configuration. <ul style="list-style-type: none"> • Use the optional <i>port-number</i> argument to specify the port number for the configuration server. The default is 80. • Use the optional source keyword and <i>ip-address</i> argument to specify the use of an IP address as the source for Cisco Networking Services configuration agent communications. • Use the optional inventory keyword to send an inventory of the linecards and modules in the device to the Cisco Networking Services configuration engine as part of the HTTP request. <p>Note The optional encrypt keyword is available only in images that support SSL.</p>

	Command or Action	Purpose
Step 4	<p>logging cns-events [<i>severity-level</i>]</p> <p>Example:</p> <pre>Device(config)# logging cns-events 2</pre>	<p>(Optional) Enables XML-formatted system event message logging to be sent through the Cisco Networking Services event bus.</p> <ul style="list-style-type: none"> Use the optional <i>severity-level</i> argument to specify the number or name of the desired severity level at which messages should be logged. The default is level 7 (debugging).
Step 5	<p>cns exec [encrypt] [<i>port-number</i>] [source {<i>ip-address</i> <i>interface-type-number</i>}]</p> <p>Example:</p> <pre>Device(config)# cns exec source 172.17.2.2</pre>	<p>(Optional) Enables and configures the Cisco Networking Services EXEC agent, which provides Cisco Networking Services EXEC services to Cisco clients.</p> <ul style="list-style-type: none"> Use the optional <i>port-number</i> argument to specify the port number for the EXEC server. The default is 80. Use the optional source keyword and <i>ip-address/interface-type number</i> argument to specify the use of an IP address as the source for Cisco Networking Services EXEC agent communications. <p>Note The optional encrypt keyword is available only in images that support SSL.</p>
Step 6	<p>cns event {<i>hostname</i> <i>ip-address</i>} [encrypt] [<i>port-number</i>] [backup] [failover-time <i>seconds</i>] [keepalive <i>seconds</i> <i>retry-count</i>] [source <i>ip-address</i> <i>interface-name</i>][clock-timeout <i>time</i>] [reconnect-time <i>time</i>]</p> <p>Example:</p> <pre>Device(config)# cns event 172.28.129.22 source 172.22.2.1</pre>	<p>Configures the Cisco Networking Services event gateway, which provides Cisco Networking Services event services to Cisco clients.</p> <ul style="list-style-type: none"> The optional encrypt keyword is available only in images that support SSL. Use the optional <i>port-number</i> argument to specify the port number for the event server. The default is 11011 with no encryption and 11012 with encryption. Use the optional backup keyword to indicate that this is the backup gateway. Before configuring a backup gateway, ensure that a primary gateway is configured. Use the optional failover-time keyword and <i>seconds</i> argument to specify a time interval in seconds to wait for the primary gateway route after the route to the backup gateway is established. Use the optional keepalive keyword with the <i>seconds</i> and <i>retry-count</i> arguments to specify the keepalive timeout in seconds and the retry count. Use the optional source keyword and <i>ip-address/interface-name</i> argument to specify the use of an IP address as the source for Cisco Networking Services event agent communications. Use the optional clock-timeout keyword to specify the maximum time, in minutes, that the Cisco Networking Services event agent will wait for the clock to be set for transports (such as SSL) that require an accurate clock. Use the optional reconnect-time keyword to specify the configurable upper limit of the maximum retry timeout. <p>Note Until the cns event command is entered, no transport connections to the Cisco Networking Services event bus are made and therefore no other Cisco Networking Services agents are operational.</p>

	Command or Action	Purpose
Step 7	exit Example: Device(config)# exit	Exits global configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

- Use the **show cns event connections** command to check that the Cisco Networking Services event agent is connected to the Cisco Networking Services event gateway.
- Use the **show cns event subject** command to check that the image agent subject names are registered. Subject names for the Cisco Networking Services image agent begin with cisco.mgmt.cns.image.

Configuration Examples for CNS Configuration Agent

Example: Enabling and Configuring Cisco Networking Services Agents

The following example shows various Cisco Networking Services agents being enabled and configured starting with the configuration agent being enabled with the **cns config partial** command to configure an incremental (partial) configuration on a remote device. The Cisco Networking Services configuration engine IP address is 172.28.129.22, and the port number is 80. The Cisco Networking Services exec agent is enabled with an IP address of 172.28.129.23, and the Cisco Networking Services event agent is enabled with an IP address of 172.28.129.24. Until the Cisco Networking Services event agent is enabled, no other Cisco Networking Services agents are operational.

```
cns config partial 172.28.129.22 80
cns exec 172.28.129.23 source 172.22.2.2
cns event 172.28.129.24 source 172.22.2.1
exit
```

In the following example, the Cisco Networking Services image agent parameters are configured using the CLI. An image ID is specified to use the IP address of the GigabitEthernet interface 0/1/1, a password is configured for the Cisco Networking Services image agent services, the Cisco Networking Services image upgrade retry interval is set to four minutes, and image management and status servers are configured.

```
cns id GigabitEthernet0/1/1 ipaddress image
cns image retry 240
cns image password abctext
cns image server https://10.21.2.3/cns/imgsvr status https://10.21.2.3/cns/status/
```

In the following example, the Cisco Networking Services image agent is configured to use the Cisco Networking Services Event Bus. An image ID is specified as the hardware serial number of the networking device, the Cisco Networking Services event agent is enabled with a number of parameters, and the Cisco Networking

Services image agent is enabled without any keywords or options. The Cisco Networking Services image agent will listen for events on the Cisco Networking Services Event Bus.

```
cns id hardware-serial image
cns event 10.21.9.7 11011 keepalive 240 120 failover-time 5
cns image
cns image password abctext
```

Example: Retrieving a Cisco Networking Services Image from a Server

In the following example, the Cisco Networking Services image agent polls a file server using the **cns image retrieve** command. Assuming that the Cisco Networking Services image agent is already enabled, the file server and status server paths specified here will overwrite any existing image agent server and status configuration. The new file server will be polled and a new image, if it exists, will be downloaded to the networking device.

```
cns image retrieve server https://10.19.2.3/cns/ status https://10.19.2.3/cnsstatus/
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco Networking Services commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples.	Cisco IOS Cisco Networking Services Command Reference
Cisco Networking Services Configuration Engine	Cisco CNS Configuration Engine Administrator Guide, 1.3

Standards and RFCs

Standard/RFC	Title
No new or modified standards/RFCs are supported by this feature, and support for existing standards/RFCs has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for CNS Configuration Agent

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 3: Feature Information for CNS Configuration Agent

Feature Name	Releases	Feature Information
CNS Configuration Agent	Cisco IOS XE Release 2.1 12.0(18)ST 12.0(22)S 12.2(2)T 12.2(8)T 12.2(33)SRA 12.2(33)SB 12.2(33)SXI	<p>The Cisco Networking Services Configuration Agent feature supports routing devices by providing the following:</p> <ul style="list-style-type: none"> • Initial configurations • Incremental (partial) configurations • Synchronized configuration updates <p>The following commands were introduced or modified by this feature: cns config cancel, cns config initial , cns config partial , cns config retrieve , cns password, debug cns config, debug cns xml-parser , show cns config outstanding ,show cns config stats, show cns config status .</p>



Cisco Networking Services Config Retrieve Enhancement with Retry and Interval

- [Finding Feature Information, page 31](#)
- [Information About CNS Config Retrieve Enhancement with Retry and Interval, page 31](#)
- [How to Configure CNS Config Retrieve Enhancement with Retry and Interval, page 32](#)
- [Configuration Examples for CNS Config Retrieve Enhancement with Retry and Interval, page 33](#)
- [Additional References, page 34](#)
- [Feature Information for CNS Config Retrieve Enhancement with Retry and Interval, page 35](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About CNS Config Retrieve Enhancement with Retry and Interval

Cisco Networking Services Config Retrieve Enhancement with Retry and Interval

The Cisco Networking Services Config Retrieve Enhancement with Retry and Interval feature adds new functionality to the **cns config retrieve** command enabling you to specify the retry interval and an amount of time in seconds to wait before attempting to retrieve a configuration from a trusted server.

How to Configure CNS Config Retrieve Enhancement with Retry and Interval

Retrieving a Cisco Networking Services Configuration from a Server

Use this task to request the configuration of a device from a configuration server. Use the **cns trusted-server** command to specify which configuration server can be used (trusted).

Before You Begin

This task assumes that you have specified a trusted server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cns config retrieve** *{host-name | ip-address}* **[encrypt]** *[port-number]* **[page page]** **[overwrite-startup]** **[retry retries interval seconds]** **[syntax-check]** **[no-persist]** *[source interface name]* **[status url]** **[event]** **[inventory]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	cns config retrieve <i>{host-name ip-address}</i> [encrypt] <i>[port-number]</i> [page page] [overwrite-startup] [retry retries interval seconds] [syntax-check] [no-persist] <i>[source interface name]</i> [status url] [event] [inventory] Example: Device(config)# cns config retrieve server1 retry 5 interval 45	Allows the device to retrieve configuration data from a web server. <ul style="list-style-type: none"> • The retry keyword is a number in the range 1 to 100, and will prompt for an interval in the range 1 to 3600 seconds. <p>Note Troubleshooting Tips If you need to stop the retrieval process, enter the Ctrl+Shift+6 key sequence.</p>

Configuration Examples for CNS Config Retrieve Enhancement with Retry and Interval

Example: Retrieving a Cisco Networking Services Configuration from a Server

Retrieving Configuration Data from the Cisco Networking Services Trusted Server

The following example shows how to request a configuration from a trusted server at 10.1.1.1:

```
cns trusted-server all 10.1.1.1
exit
```

```
cns config retrieve 10.1.1.1
```

The following example shows how to request a configuration from a trusted server at 10.1.1.1 and to configure a Cisco Networking Services configuration retrieve interval using the **cns config retrieve** command:

```
cns trusted-server all 10.1.1.1
exit
```

```
cns config retrieve 10.1.1.1 retry 50 interval 1500
```

```
CNS Config Retrieve Attempt 1 out of 50 is in progress
```

```
Next cns config retrieve retry is in 1499 seconds (Ctrl-Shift-6 to abort this command).
```

```
..
```

```
00:26:40: %CNS-3-TRANSPORT: CNS_HTTP_CONNECTION_FAILED:10.1.1.1 -Process= "CNS config retv",
ipl= 0, pid= 43
```

```
00:26:40: %CNS-3-TRANSPORT: CNS_HTTP_CONNECTION_FAILED -Process= "CNS config retv",
ipl= 0, pid= 43.....
```

```
cns config retrieve 10.1.1.1
```

Applying the Retrieved Data to the Running Configuration File

The following example shows how to check and apply configuration data retrieved from the server to running configuration file only. The Cisco Networking Services Configuration Agent will attempt to retrieve configuration data at 30-second intervals until the attempt is successful, or is unsuccessful five times in these attempts.

```
cns config retrieve 10.1.1.1 syntax-check no-persist retry 5 interval 30
```

Overwriting the Startup Configuration File with the Retrieved Data

The following example shows how to overwrite the startup configuration file with the configuration data retrieved from the server. The configuration data will not be applied to the running configuration.

```
cns config retrieve 10.1.1.1 syntax-check no-persist retry 5 interval 30
```

```
cns config retrieve 10.1.1.1 overwrite-startup
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco Networking Services commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples.	Cisco IOS Cisco Networking Services Command Reference
Cisco Networking Services Configuration Engine	Cisco CNS Configuration Engine Administrator Guide, 1.3

Standards and RFCs

Standard/RFC	Title
No new or modified standards/RFCs are supported by this feature, and support for existing standards/RFCs has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for CNS Config Retrieve Enhancement with Retry and Interval

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 4: Feature Information for Cisco Networking Services Config Retrieve Enhancement with Retry and Interval

Feature Name	Releases	Feature Information
Cisco Networking Services Config Retrieve Enhancement with Retry and Interval	Cisco IOS XE Release 2.1 12.4(15)T 12.2(33)SRC 12.2(33)SB 12.2(50)SY	The Cisco Networking Services Config Retrieve Enhancement with Retry and Interval feature adds two options to the cns config retrieve command enabling you to specify an amount of time in seconds to wait before attempting to retrieve a configuration from a trusted server. The number of retries is restricted to 100 to prevent the configuration agent from indefinitely attempting to reach an unreachable server. Use the keyboard combination Ctrl-Shift-6 to abort the cns config retrieve command. The following command was modified by this feature: cns config retrieve .



Cisco Networking Services Interactive CLI

- [Finding Feature Information, page 37](#)
- [Information About CNS Interactive CLI, page 37](#)
- [Additional References, page 38](#)
- [Feature Information for CNS Interactive CLI, page 38](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About CNS Interactive CLI

Cisco Networking Services Interactive CLI

The Cisco Networking Services Interactive CLI feature provides a XML interface that allows you to send interactive commands to a device, such as commands that generate prompts for user input. A benefit of this feature is that interactive commands can be aborted before they have been fully processed. For example, for commands that generate a significant amount of output, the XML interface can be customized to limit the size of the output or the length of time allowed for the output to accumulate. The capability to use a programmable interface to abort a command before its normal termination (similar to manually aborting a command) can greatly increase the efficiency of diagnostic applications that might use this functionality. The new XML interface also allows for multiple commands to be processed in a single session. The response for each command is packaged together and sent in a single response event.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco Networking Services commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS Cisco Networking Services Command Reference
Cisco Networking Services Configuration Engine	Cisco CNS Configuration Engine Administrator Guide, 1.3

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for CNS Interactive CLI

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 5: Feature Information for Cisco Networking Services Interactive CLI

Feature Name	Releases	Feature Information
Cisco Networking Services Interactive CLI	Cisco IOS XE Release 2.1 12.0(28)S 12.2(18)SXE 12.2(18)SXF2 12.2(33)SRC 12.2(33)SXI	The Cisco Networking Services Interactive CLI feature introduces an XML interface that allows you to send interactive commands to a device, such as commands that generate prompts for user input.



Command Scheduler (Kron)

- [Finding Feature Information](#), page 41
- [Restrictions for Command Scheduler](#), page 41
- [Information About Command Scheduler \(Kron\)](#), page 42
- [How to Configure Command Scheduler \(Kron\)](#), page 42
- [Configuration Examples for Command Scheduler \(Kron\)](#), page 45
- [Additional References](#), page 46
- [Feature Information for Command Scheduler \(Kron\)](#), page 47

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for Command Scheduler

The EXEC CLI specified in a Command Scheduler policy list must neither generate a prompt nor can it be terminated using keystrokes. Command Scheduler is designed as a fully automated facility, and no manual intervention is permitted.

Information About Command Scheduler (Kron)

Command Scheduler

The Command Scheduler (KRON) Policy for System Startup feature enables support for the Command Scheduler upon system startup.

The Command Scheduler allows customers to schedule fully-qualified EXEC mode CLI commands to run once, at specified intervals, at specified calendar dates and times, or upon system startup. Originally designed to work with Cisco Networking Services commands, Command Scheduler now has a broader application. Using the Cisco Networking Services image agent feature, remote devices residing outside a firewall or using Network Address Translation (NAT) addresses can use Command Scheduler to launch CLI at intervals, to update the image running in the device.

Command Scheduler has two basic processes. A policy list is configured containing lines of fully-qualified EXEC CLI commands to be run at the same time or same interval. One or more policy lists are then scheduled to run after a specified interval of time, at a specified calendar date and time, or upon system startup. Each scheduled occurrence can be set to run either once only or on a recurring basis.

How to Configure Command Scheduler (Kron)

Configuring Command Scheduler Policy Lists and Occurrences

An occurrence for Command Scheduler is defined as a scheduled event. Policy lists are configured to run after a specified interval of time, at a specified calendar date and time, or upon system startup. Policy lists can be run once, as a one-time event, or as recurring events over time.

Command Scheduler occurrences can be scheduled before the associated policy list has been configured, but a warning will advise you to configure the policy list before it is scheduled to run.

Before You Begin

Perform this task to set up Command Scheduler policy lists of EXEC Cisco Networking Services commands and configure a Command Scheduler occurrence to specify the time or interval after which the Cisco Networking Services commands will run.

Command Scheduler Policy Lists

Policy lists consist of one or more lines of fully-qualified EXEC CLI commands. All commands in a policy list are executed when the policy list is run by Command Scheduler using the **kron occurrence** command. Use separate policy lists for CLI commands that are run at different times. No editor function is available, and the policy list is run in the order in which it was configured. To delete an entry, use the **no** form of the **cli** command followed by the appropriate EXEC command. If an existing policy list name is used, new entries are added to the end of the policy list. To view entries in a policy list, use the **show running-config** command. If a policy list is scheduled to run only once, it will not be displayed by the **show running-config** command after it has run.

Policy lists can be configured after the policy list has been scheduled, but each policy list must be configured before it is scheduled to run.

Command Scheduler Occurrences

The clock time must be set on the routing device before a Command Scheduler occurrence is scheduled to run. If the clock time is not set, a warning message will appear on the console screen after the **kron occurrence** command has been entered. Use the **clock** command or Network Time Protocol (NTP) to set the clock time.

The EXEC CLI to be run by Command Scheduler must be tested on the routing device to determine if it will run without generating a prompt or allowing execution interruption by keystrokes. Initial testing is important because Command Scheduler will delete the entire policy list if any CLI syntax fails. Removing the policy list ensures that any CLI dependencies will not generate more errors.

If you use the **conditional** keyword with the **kron policy-list** command, execution of the commands will stop when an error is encountered.

**Note**

- No more than 31 policy lists can be scheduled to run at the same time.
- If a one-time occurrence is scheduled, the occurrence will not be displayed by the **show running-config** command after the occurrence has run.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **kron policy-list** *list-name* [**conditional**]
4. **cli** *command*
5. **exit**
6. **kron occurrence** *occurrence-name* [**user** *username*] {**in**[[*numdays*:]*numhours*:]*nummin*| **at** *hours:min*[[*month*] *day-of-month*] [*day-of-week*]} {**oneshot**| **recurring**| **system-startup**}
7. **policy-list** *list-name*
8. **exit**
9. **show kron schedule**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	<p>kron policy-list <i>list-name</i> [conditional]</p> <p>Example:</p> <pre>Device(config)# kron policy-list cns-weekly</pre>	<p>Specifies a name for a new or existing Command Scheduler policy list and enters kron-policy configuration mode.</p> <ul style="list-style-type: none"> • If the <i>list-name</i> is new, a new policy list structure is created. • If the <i>list-name</i> exists, the existing policy list structure is accessed. The policy list is run in configured order with no editor function. • If the optional conditional keyword is used, execution of the commands stops when an error is encountered.
Step 4	<p>cli <i>command</i></p> <p>Example:</p> <pre>Device(config-kron-policy)# cli cns image retrieve server https://10.19.2.3/cnsweek/ status https://10.19.2.3/cnsstatus/week/</pre>	<p>Specifies the fully-qualified EXEC command and associated syntax to be added as an entry in the specified Command Scheduler policy list.</p> <ul style="list-style-type: none"> • Each entry is added to the policy list in the order in which it is configured. • Repeat this step to add other EXEC CLI commands to a policy list to be executed at the same time or interval. <p>Note EXEC commands that generate a prompt or can be terminated using keystrokes will cause an error.</p>
Step 5	<p>exit</p> <p>Example:</p> <pre>Device(config-kron-policy)# exit</pre>	<p>Exits kron-policy configuration mode and returns the device to global configuration mode.</p>
Step 6	<p>kron occurrence <i>occurrence-name</i> [user <i>username</i>]</p> <p>{in[[<i>numdays:</i>]<i>numhours:</i>]<i>nummin</i> at <i>hours:min</i>[[<i>month</i>] <i>day-of-month</i>] [<i>day-of-week</i>]} {oneshot recurring system-startup}</p> <p>Example:</p> <pre>Device(config)# kron occurrence may user sales at 6:30 may 20 oneshot</pre>	<p>Specifies a name and schedule for a new or existing Command Scheduler occurrence and enters kron-occurrence configuration mode.</p> <ul style="list-style-type: none"> • Use the in keyword to specify a delta time interval with a timer that starts when this command is configured. • Use the at keyword to specify a calendar date and time. • Choose either the oneshot or recurring keyword to schedule Command Scheduler occurrence once or repeatedly. Add the optional system-startup keyword for the occurrence to be at system startup.
Step 7	<p>policy-list <i>list-name</i></p> <p>Example:</p> <pre>Device(config-kron-occurrence)# policy-list sales-may</pre>	<p>Specifies a Command Scheduler policy list.</p> <ul style="list-style-type: none"> • Each entry is added to the occurrence list in the order in which it is configured. <p>Note If the CLI commands in a policy list generate a prompt or can be terminated using keystrokes, an error will be generated and the policy list will be deleted.</p>

	Command or Action	Purpose
Step 8	exit Example: Device(config-kron-occurrence)# exit	Exits kron-occurrence configuration mode and returns the device to global configuration mode. <ul style="list-style-type: none"> Repeat this step to exit global configuration mode.
Step 9	show kron schedule Example: Device# show kron schedule	(Optional) Displays the status and schedule information of Command Scheduler occurrences.

Examples

In the following example, output information is displayed about the status and schedule of all configured Command Scheduler occurrences:

```
Device# show kron schedule
Kron Occurrence Schedule
cns-weekly inactive, will run again in 7 days 01:02:33
may inactive, will run once in 32 days 20:43:31 at 6:30 on May 20
```

Troubleshooting Tips

Use the **debug kron** command in privileged EXEC mode to troubleshoot Command Scheduler command operations. Use any debugging command with caution because the volume of output generated can slow or stop the device's operations.

Configuration Examples for Command Scheduler (Kron)

Example: Command Scheduler Policy Lists and Occurrences

In the following example, a Command Scheduler policy named cns-weekly is configured to run two sets of EXEC CLI involving Cisco Networking Services commands. The policy is then scheduled with two other policies to run every seven days, one hour and thirty minutes.

```
kron policy-list cns-weekly
cli cns image retrieve server http://10.19.2.3/week/ status http://10.19.2.5/status/week/
cli cns config retrieve page /testconfig/config.asp no-persist
exit
kron occurrence week in 7:1:30 recurring
policy-list cns-weekly
policy-list itd-weekly
policy-list mkt-weekly
```

In the following example, a Command Scheduler policy named sales-may is configured to run a Cisco Networking Services command to retrieve a specified image from a remote server. The policy is then scheduled to run only once on May 20, at 6:30 a.m.

```
kron policy-list sales-may
cli cns image retrieve server 10.19.2.3 status 10.19.2.3
exit
kron occurrence may at 6:30 May 20 oneshot
policy-list sales-may
```

In the following example, a Command Scheduler policy named image-sunday is configured to run a Cisco Networking Services command to retrieve a specified image from a remote server. The policy is then scheduled to run every Sunday at 7:30 a.m.

```
kron policy-list image-sunday
cli cns image retrieve server 10.19.2.3 status 10.19.2.3
exit
kron occurrence sunday user sales at 7:30 sunday recurring
policy-list image-sunday
```

In the following example, a Command Scheduler policy named file-retrieval is configured to run a Cisco Networking Services command to retrieve a specific file from a remote server. The policy is then scheduled to run on system startup.

```
kron policy-list file-retrieval
cli cns image retrieve server 10.19.2.3 status 10.19.2.3
exit
kron occurrence system-startup
policy-list file-retrieval
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco Networking Services commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples.	Cisco IOS Cisco Networking Services Command Reference
Cisco Networking Services Configuration Engine	Cisco CNS Configuration Engine Administrator Guide, 1.3

Standards and RFCs

Standard/RFC	Title
No new or modified standards/RFCs are supported by this feature, and support for existing standards/RFCs has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Command Scheduler (Kron)

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 6: Feature Information for Command Scheduler (Kron)

Feature Name	Releases	Feature Information
Command Scheduler (Kron)	Cisco IOS XE Release 2.1 12.3(1) 12.2(33)SRA 12.2(33)SRC 12.2(33)SB 12.2(33)SXI 12.2(50)SY	The Command Scheduler feature provides the ability to schedule some EXEC CLI commands to run at specific times or at specified intervals. The following commands were introduced or modified by this feature: cli , debug kron , kron occurrence , kron policy-list , policy-list , show kron schedule .

Feature Name	Releases	Feature Information
Command Scheduler (Kron) Policy for System Startup	12.2(33)SRC 12.2(50)SY 12.2(33)SB 12.4(15)T	The Command Scheduler (Kron) Policy for System Startup feature enables support for the Command Scheduler feature upon system startup.



Network Configuration Protocol

The Network Configuration Protocol (NETCONF) defines a simple mechanism through which a network device can be managed, configuration data can be retrieved, and new configuration data can be uploaded and manipulated. NETCONF uses Extensible Markup Language (XML)-based data encoding for the configuration data and protocol messages.

- [Finding Feature Information](#), page 49
- [Prerequisites for NETCONF](#), page 49
- [Information About NETCONF](#), page 50
- [How to Configure NETCONF](#), page 50
- [Configuration Examples for NETCONF](#), page 57
- [Additional References for NETCONF](#), page 60
- [Feature Information for NETCONF](#), page 61
- [Glossary](#), page 62

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for NETCONF

A vty line must be available for each NETCONF session as specified by the **netconf max-session** command.

Information About NETCONF

NETCONF Notifications

NETCONF sends notifications of any configuration change over NETCONF. A notification is an event indicating that a configuration change has occurred. The change can be a new configuration, deleted configuration, or changed configuration. The notifications are sent at the end of a successful configuration operation as one message that shows the set of changes rather than showing individual messages for each line that is changed in the configuration.

How to Configure NETCONF

Configuring the NETCONF Network Manager Application

Step 1 Use the following CLI string to configure the NETCONF network manager application to invoke NETCONF as an SSH subsystem:

Example:

```
Unix Side: ssh-2 -s companyname@10.1.1.1 netconf
```

Step 2 As soon as the NETCONF session is established, indicate the server capabilities by sending an XML document containing a <hello>:

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
  <hello>
    <capabilities>
      <capability>
        urn:ietf:params:xml:ns:netconf:base:1.0
      </capability>
      <capability>
        urn:ietf:params:ns:netconf:capability:startup:1.0
      </capability>
    </capabilities>
    <session-id>4<session-id>
  </hello>]]>]]>
```

The client also responds by sending an XML document containing a <hello>:

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
  <hello>
    <capabilities>
      <capability>
        urn:ietf:params:xml:ns:netconf:base:1.0
      </capability>
```



```

    </capabilities>
</hello>]]>]]>

```

Note Although the example shows the server sending a <hello> message followed by the message from the client, both sides send the message as soon as the NETCONF subsystem is initialized, perhaps simultaneously.

Tip All NETCONF requests must end with]]>]]> which denotes an end to the request. Until the]]>]]> sequence is sent, the device will not process the request.

See the “Example: Configuring NETCONF over SSHv2” section for a specific example.

Step 3 Use the following XML string to enable the NETCONF network manager application to send and receive NETCONF notifications:

Example:

```

<?xml version="1.0" encoding="UTF-8" ?>
<rpc message-id="9.0"><notification-on/>
</rpc>]]>]]>

```

Step 4 Use the following XML string to stop the NETCONF network manager application from sending or receiving NETCONF notifications:

Example:

```

<?xml version="1.0" encoding="UTF-8" ?>
<rpc message-id="9.13"><notification-off/>
</rpc>]]>]]>

```

Delivering NETCONF Payloads

Use the following XML string to deliver the NETCONF payload to the network manager application:

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.cisco.com/cpi_10/schema" elementFormDefault="qualified"
  attributeFormDefault="unqualified" xmlns="http://www.cisco.com/cpi_10/schema"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <!--The following elements define the cisco extensions for the content of the filter
  element in a <get-config> request. They allow the client to specify the format of the
  response and to select subsets of the entire configuration to be included.-->
  <xs:element name="config-format-text-block">
    <xs:annotation>
      <xs:documentation>If this element appears in the filter, then the client is
      requesting that the response data be sent in config command block format.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="text-filter-spec" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="config-format-text-cmd">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="text-filter-spec"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="config-format-xml">

```

```

    <xs:annotation>
      <xs:documentation>When this element appears in the filter of a get-config request,
the results are to be returned in E-DI XML format. The content of this element is treated
as a filter.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:complexContent>
        <xs:extension base="xs:anyType"/>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
  <!--These elements are used in the filter of a <get> to specify operational data to
return.-->
  <xs:element name="oper-data-format-text-block">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="show" type="xs:string" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="oper-data-format-xml">
    <xs:complexType>
      <xs:sequence>
        <xs:any/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <!--When config-format-text format is specified, the following describes the content
of the data element in the response-->
  <xs:element name="cli-config-data">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="cmd" type="xs:string" maxOccurs="unbounded">
          <xs:annotation>
            <xs:documentation>Content is a command. May be multiple
lines.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="cli-config-data-block" type="xs:string">
    <xs:annotation>
      <xs:documentation>The content of this element is the device configuration as it
would be sent to a terminal session. It contains embedded newline characters that must be
preserved as they represent the boundaries between the individual command
lines</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="text-filter-spec">
    <xs:annotation>
      <xs:documentation>If this element is included in the config-format-text element,
then the content is treated as if the string was appended to the "show running-config"
command line.</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="cli-oper-data-block">
    <xs:complexType>
      <xs:annotation>
        <xs:documentation> This element is included in the response to get operation.
Content of this element is the operational data in text format.</xs:documentation>
      </xs:annotation>
      <xs:sequence>
        <xs:element name="item" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="exec"/>
              <xs:element name="show"/>
              <xs:element name="response"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```

```

    </xs:complexType>
  </xs:element>
</xs:schema>

```

Formatting NETCONF Notifications

The NETCONF network manager application uses .xsd schema files to describe the format of the XML NETCONF notification messages that are sent between a NETCONF network manager application and a device running NETCONF over SSHv2 or BEEP. These files can be displayed in a browser or a schema reading tool. You can use these schemas to validate that the XML is correct. These schemas describe the format, not the content, of the data being exchanged.

NETCONF uses the <edit-config> function to load all of a specified configuration to a specified target configuration. When this new configuration is entered, the target configuration is not replaced. The target configuration is changed according to the data and requested operations of the requesting source.

The following are schemas for the NETCONF <edit-config> function in CLI, CLI block, and XML format.

NETCONF <edit-config> Request: CLI Format

```

<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <cli-config-data>
<cmd>hostname test</cmd>
      <cmd>interface fastEthernet0/1</cmd>
      <cmd>ip address 192.168.1.1 255.255.255.0</cmd>
</cli-config-data>
    </config>
  </edit-config>
</rpc>]]>]]>

```

NETCONF <edit-config> Response: CLI Format

```

<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="101" xmlns="urn:ietf:params:netconf:base:1.0">
  <ok/>
</rpc-reply>]]>]]>

```

NETCONF <edit-config> Request: CLI-Block Format

```

<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="netconf.mini.edit.3">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <cli-config-data-block>
        hostname bob
        interface fastEthernet0/1
        ip address 192.168.1.1 255.255.255.0
      </cli-config-data-block>
    </config>
  </edit-config>
</rpc>]]>]]>

```

NETCONF <edit-config> Response: CLI-Block Format

```
<?xml version="1.0" encoding="\UTF-8\"?>
<rpc-reply message-id="netconf.mini.edit.3" xmlns="urn:ietf:params:netconf:base:1.0">
  <ok/>
</rpc-reply>]]]]>
```

The following are schemas for the NETCONF <get-config> function in CLI and CLI-block format.

NETCONF <get-config> Request: CLI Format

```
<?xml version="1.0" encoding="\UTF-8\"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-config>
    <source>
      <running/>
    </source>
    <filter>
      <config-format-text-cmd>
        <text-filter-spec> | inc interface </text-filter-spec>
      </config-format-text-cmd>
    </filter>
  </get-config>
</rpc>]]]]>
```

NETCONF <get-config> Response: CLI Format

```
<?xml version="1.0" encoding="\UTF-8\"?>
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <cli-config-data>
      <cmd>interface FastEthernet0/1</cmd>
      <cmd>interface FastEthernet0/2</cmd>
    </cli-config-data>
  </data>
</rpc-reply>]]]]>
```

NETCONF <get-config> Request: CLI-Block Format

```
<?xml version="1.0" encoding="\UTF-8\"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-config>
    <source>
      <running/>
    </source>
    <filter>
      <config-format-text-block>
        <text-filter-spec> | inc interface </text-filter-spec>
      </config-format-text-block>
    </filter>
  </get-config>
</rpc>]]]]>
```

NETCONF <get-config> Response: CLI-Block Format

```
<?xml version="1.0" encoding="\UTF-8\"?>
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <cli-config-data-block>
      interface FastEthernet0/1
      interface FastEthernet0/2
    </cli-config-data-block>
  </data>
</rpc-reply>]]]]>
```

NETCONF uses the <get> function to retrieve configuration and device-state information. The NETCONF <get> format is the equivalent of a Cisco IOS **show** command. The <filter> parameter specifies the portion of the system configuration and device-state data to retrieve. If the <filter> parameter is empty, nothing is returned.

The following are schemas for the <get> function in CLI and CLI-block format.

NETCONF <get> Request: CLI Format

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get>
    <filter>
      <config-format-text-cmd>
        <text-filter-spec> | include interface </text-filter-spec>
      </config-format-text-cmd>
      <oper-data-format-text-block>
        <exec>show interfaces</exec>
        <exec>show arp</exec>
      </oper-data-format-text-block>
    </filter>
  </get>
</rpc>]]]]>
```

NETCONF <get> Response: CLI Format

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <cli-config-data>
      <cmd>interface Loopback0</cmd>
      <cmd>interface GigabitEthernet0/1</cmd>
      <cmd>interface GigabitEthernet0/2</cmd>
    </cli-config-data>
    <cli-oper-data-block>
      <item>
        <exec>show interfaces</exec>
        <response>
          <!-- output of "show interfaces" ----->
        </response>
      </item>
      <item>
        <exec>show arp</exec>
        <response>
          <!-- output of "show arp" ----->
        </response>
      </item>
    </cli-oper-data-block>
  </data>
</rpc-reply>]]]]>
```

NETCONF <get> Request: CLI-Block Format

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get>
    <filter>
      <config-format-text-block>
        <text-filter-spec> | include interface </text-filter-spec>
      </config-format-text-block>
      <oper-data-format-text-block>
        <exec>show interfaces</exec>
        <exec>show arp</exec>
      </oper-data-format-text-block>
    </filter>
```

```

</get>
</rpc>]]>]]>

```

NETCONF <get> Response: CLI-Block Format

```

<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <cli-config-data-block>
      interface Loopback0
      interface GigabitEthernet0/1
      interface GigabitEthernet0/2
    </cli-config-data-block>
    <cli-oper-data-block>
      <item>
        <exec>show interfaces</exec>
        <response>
          <!-- output of "show interfaces" ----->
        </response>
      </item>
      <item>
        <exec>show arp</exec>
        <response>
          <!-- output of "show arp" ----->
        </response>
      </item>
    </cli-oper-data-block>
  </data>
</rpc-reply>]]>]]>

```

Monitoring and Maintaining NETCONF Sessions



Note

- A minimum of four concurrent NETCONF sessions must be configured.
- A maximum of 16 concurrent NETCONF sessions can be configured.
- NETCONF does not support SSHv1.

SUMMARY STEPS

1. **enable**
2. **show netconf {counters | session| schema}**
3. **debug netconf {all | error}**
4. **clear netconf {counters | sessions}**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	show netconf {counters session schema} Example: Device# show netconf counters	Displays NETCONF information.
Step 3	debug netconf {all error} Example: Device# debug netconf error	Enables debugging of NETCONF sessions.
Step 4	clear netconf {counters sessions} Example: Device# clear netconf sessions	Clears NETCONF statistics counters and NETCONF sessions, and frees associated resources and locks.

Configuration Examples for NETCONF

Example: Configuring the NETCONF Network Manager Application

The following example shows how to configure the NETCONF network manager application to invoke NETCONF as an SSH subsystem:

```
Unix Side: ssh-2 -s companyname@10.1.1.1 netconf
```

As soon as the NETCONF session is established, indicate the server capabilities by sending an XML document containing a <hello>:

```
<?xml version="1.0" encoding="UTF-8"?>
  <hello>
    <capabilities>
      <capability>
        urn:ietf:params:xml:ns:netconf:base:1.0
      </capability>
      <capability>
        urn:ietf:params:ns:netconf:capability:startup:1.0
      </capability>
    </capabilities>
    <session-id>4</session-id>
  </hello>]]>]]>
```

The client also responds by sending an XML document containing a <hello>:

```
<?xml version="1.0" encoding="UTF-8"?>
  <hello>
    <capabilities>
      <capability>
        urn:ietf:params:xml:ns:netconf:base:1.0
```

```

    </capability>
  </capabilities>
</hello>]]>]]>

```

Use the following XML string to enable the NETCONF network manager application to send and receive NETCONF notifications:

```

<?xml version="1.0" encoding="UTF-8" ?>
<rpc message-id="9.0"><notification-on/>
</rpc>]]>]]>

```

Use the following XML string to stop the NETCONF network manager application from sending or receiving NETCONF notifications:

```

<?xml version="1.0" encoding="UTF-8" ?>
<rpc message-id="9.13"><notification-off/>
</rpc>]]>]]>

```

Example: Monitoring NETCONF Sessions

The following is sample output from the **show netconf counters** command:

```

Device# show netconf counters
NETCONF Counters
Connection Attempts:0: rejected:0 no-hello:0 success:0
Transactions
  total:0, success:0, errors:0
detailed errors:
  in-use 0          invalid-value 0          too-big 0
  missing-attribute 0      bad-attribute 0          unknown-attribute 0
  missing-element 0      bad-element 0          unknown-element 0
  unknown-namespace 0    access-denied 0          lock-denied 0
  resource-denied 0      rollback-failed 0        data-exists 0
  data-missing 0         operation-not-supported 0  operation-failed 0
  partial-operation 0

```

The following is sample output from the **show netconf session** command:

```

Device# show netconf session
(Current | max) sessions: 3 | 4
Operations received: 100          Operation errors: 99
Connection Requests: 5           Authentication errors: 2   Connection Failures: 0
ACL dropped : 30
Notifications Sent: 20

```

The output of the **show netconf schema** command displays the element structure for a NETCONF request and the resulting reply. This schema can be used to construct proper NETCONF requests and parse the resulting replies. The nodes in the schema are defined in RFC 4741. The following is sample output from the **show netconf schema** command:

```

Device# show netconf schema
New Name Space 'urn:ietf:params:xml:ns:netconf:base:1.0'
<VirtualRootTag> [0, 1] required
  <rpc-reply> [0, 1] required
    <ok> [0, 1] required
    <data> [0, 1] required
    <rpc-error> [0, 1] required
      <error-type> [0, 1] required
      <error-tag> [0, 1] required
      <error-severity> [0, 1] required
      <error-app-tag> [0, 1] required
      <error-path> [0, 1] required
      <error-message> [0, 1] required
      <error-info> [0, 1] required

```



```

    <bad-attribute> [0, 1] required
    <bad-element> [0, 1] required
    <ok-element> [0, 1] required
    <err-element> [0, 1] required
    <noop-element> [0, 1] required
    <bad-namespace> [0, 1] required
    <session-id> [0, 1] required
<hello> [0, 1] required
  <capabilities> 1 required
  <capability> 1+ required
<rpc> [0, 1] required
  <close-session> [0, 1] required
  <commit> [0, 1] required
  <confirmed> [0, 1] required
  <confirm-timeout> [0, 1] required
  <copy-config> [0, 1] required
  <source> 1 required
  <config> [0, 1] required
  <cli-config-data> [0, 1] required
  <cmd> 1+ required
  <cli-config-data-block> [0, 1] required
  <xml-config-data> [0, 1] required
  <Device-Configuration> [0, 1] required
  <> any subtree is allowed
  <candidate> [0, 1] required
  <running> [0, 1] required
  <startup> [0, 1] required
  <url> [0, 1] required
  <target> 1 required
  <candidate> [0, 1] required
  <running> [0, 1] required
  <startup> [0, 1] required
  <url> [0, 1] required
  <delete-config> [0, 1] required
  <target> 1 required
  <candidate> [0, 1] required
  <running> [0, 1] required
  <startup> [0, 1] required
  <url> [0, 1] required
  <discard-changes> [0, 1] required
  <edit-config> [0, 1] required
  <target> 1 required
  <candidate> [0, 1] required
  <running> [0, 1] required
  <startup> [0, 1] required
  <url> [0, 1] required
  <default-operation> [0, 1] required
  <test-option> [0, 1] required
  <error-option> [0, 1] required
  <config> 1 required
  <cli-config-data> [0, 1] required
  <cmd> 1+ required
  <cli-config-data-block> [0, 1] required
  <xml-config-data> [0, 1] required
  <Device-Configuration> [0, 1] required
  <> any subtree is allowed
<get> [0, 1] required
  <filter> [0, 1] required
  <config-format-text-cmd> [0, 1] required
  <text-filter-spec> [0, 1] required
  <config-format-text-block> [0, 1] required
  <text-filter-spec> [0, 1] required
  <config-format-xml> [0, 1] required
  <oper-data-format-text-block> [0, 1] required
  <exec> [0, 1] required
  <show> [0, 1] required
  <oper-data-format-xml> [0, 1] required
  <exec> [0, 1] required
  <show> [0, 1] required
  <get-config> [0, 1] required
  <source> 1 required
  <config> [0, 1] required
  <cli-config-data> [0, 1] required

```

```

    <cmd> 1+ required
    <cli-config-data-block> [0, 1] required
    <xml-config-data> [0, 1] required
    <Device-Configuration> [0, 1] required
    <> any subtree is allowed
    <candidate> [0, 1] required
    <running> [0, 1] required
    <startup> [0, 1] required
    <url> [0, 1] required
    <filter> [0, 1] required
    <config-format-text-cmd> [0, 1] required
    <text-filter-spec> [0, 1] required
    <config-format-text-block> [0, 1] required
    <text-filter-spec> [0, 1] required
    <config-format-xml> [0, 1] required
    <kill-session> [0, 1] required
    <session-id> [0, 1] required
    <lock> [0, 1] required
    <target> 1 required
    <candidate> [0, 1] required
    <running> [0, 1] required
    <startup> [0, 1] required
    <url> [0, 1] required
    <unlock> [0, 1] required
    <target> 1 required
    <candidate> [0, 1] required
    <running> [0, 1] required
    <startup> [0, 1] required
    <url> [0, 1] required
    <validate> [0, 1] required
    <source> 1 required
    <config> [0, 1] required
    <cli-config-data> [0, 1] required
    <cmd> 1+ required
    <cli-config-data-block> [0, 1] required
    <xml-config-data> [0, 1] required
    <Device-Configuration> [0, 1] required
    <> any subtree is allowed
    <candidate> [0, 1] required
    <running> [0, 1] required
    <startup> [0, 1] required
    <url> [0, 1] required
    <notification-on> [0, 1] required
    <notification-off> [0, 1] required

```

Additional References for NETCONF

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
NETCONF commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	<i>Cisco IOS Cisco Networking Services Command Reference</i>
Security and IP access lists commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	<i>Cisco IOS Security Command Reference</i>

Standards and RFCs

Standard/RFC	Title
RFC 4251	<i>The Secure Shell (SSH) Protocol Architecture</i>
RFC 4252	<i>The Secure Shell (SSH) Authentication Protocol</i>
RFC 4741	<i>NETCONF Configuration Protocol</i>
RFC 4744	<i>Using the NETCONF Protocol over the Blocks Extensible Exchange Protocol (BEEP)</i>

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for NETCONF

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 7: Feature Information for NETCONF

Feature Name	Releases	Feature Information
NETCONF		<p>The NETCONF protocol defines a simple mechanism through which a network device can be managed, configuration data can be retrieved, and new configuration data can be uploaded and manipulated. NETCONF uses Extensible Markup Language (XML)-based data encoding for the configuration data and protocol messages.</p> <p>The following commands were introduced or modified by this feature: clear netconf, debug netconf, show netconf.</p>
NETCONF XML PI		<p>The NETCONF protocol was enhanced, adding format attribute support for all Cisco IOS exec commands.</p> <p>The following commands were modified: clear netconf, debug netconf, and show netconf.</p>

Glossary

BEEP —Blocks Extensible Exchange Protocol. A generic application protocol framework for connection-oriented, asynchronous interactions.

NETCONF —Network Configuration Protocol. A protocol that defines a simple mechanism through which a network device can be managed, configuration data can be retrieved, and new configuration data can be uploaded and manipulated.

SASL —Simple Authentication and Security Layer. An Internet standard method for adding authentication support to connection-based protocols. SASL can be used between a security appliance and a Lightweight Directory Access Protocol (LDAP) server to secure user authentication.

SSHv2 —Secure Shell Version 2. SSH runs on top of a reliable transport layer and provides strong authentication and encryption capabilities. SSHv2 provides a means to securely access and securely execute commands on another computer over a network.

TLS —Transport Layer Security. An application-level protocol that provides for secure communication between a client and server by allowing mutual authentication, the use of hash for integrity, and encryption for privacy. TLS relies upon certificates, public keys, and private keys.

XML —Extensible Markup Language. A standard maintained by the World Wide Web Consortium (W3C) that defines a syntax that lets you create markup languages to specify information structures. Information

structures define the type of information (for example, subscriber name or address), not how the information appears (bold, italic, and so on). External processes can manipulate these information structures and publish them in a variety of formats. XML allows you to define your own customized markup language.



NETCONF over SSHv2

You can use the Network Configuration Protocol (NETCONF) over Secure Shell Version 2 (SSHv2) feature to perform network configurations via the Cisco command-line interface (CLI) over an encrypted transport. The NETCONF Network Manager, which is the NETCONF client, must use Secure Shell Version 2 (SSHv2) as the network transport to the NETCONF server. Multiple NETCONF clients can connect to the NETCONF server.

- [Finding Feature Information, page 65](#)
- [Prerequisites for NETCONF over SSHv2, page 65](#)
- [Restrictions for NETCONF over SSH, page 66](#)
- [Information About NETCONF over SSHv2, page 66](#)
- [How to Configure NETCONF over SSHv2, page 67](#)
- [Configuration Examples for NETCONF over SSHv2, page 73](#)
- [Additional References for NETCONF over SSHv2, page 75](#)
- [Feature Information for NETCONF over SSHv2, page 77](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for NETCONF over SSHv2

- NETCONF over SSHv2 requires that a vty line be available for each NETCONF session as specified in the `netconf max-session` command.

Restrictions for NETCONF over SSH

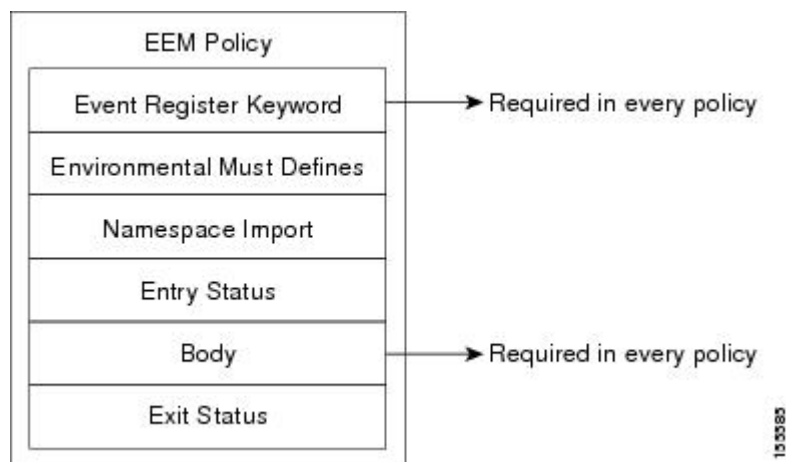
- Network Configuration Protocol (NETCONF) Secure Shell Version 2 (SSHv2) supports a maximum of 16 concurrent sessions.
- Only SSH version 2 is supported.

Information About NETCONF over SSHv2

NETCONF over SSHv2

To run the NETCONF over SSHv2 feature, the client (a Cisco device running Cisco software) establishes an SSH transport connection with the server (a NETCONF network manager). The following image shows a basic NETCONF over SSHv2 network configuration. The client and server exchange keys for security and password encryption. The user ID and password of the SSHv2 session running NETCONF are used for authorization and authentication purposes. The user privilege level is enforced and the client session may not have full access to the NETCONF operations if the privilege level is not high enough. If authentication, authorization, and accounting (AAA) is configured, the AAA service is used as if a user had established an SSH session directly to the device. Using the existing security configuration makes the transition to NETCONF almost seamless. Once the client has been successfully authenticated, the client invokes the SSH connection protocol and the SSH session is established. After the SSH session is established, the user or application invokes NETCONF as an SSH subsystem called “netconf.”

Figure 1: NETCONF over SSHv2



Secure Shell Version 2

SSHv2 runs on top of a reliable transport layer and provides strong authentication and encryption capabilities. SSHv2 provides a means to securely access and securely execute commands on another computer over a network.

NETCONF does not support SSH version 1. The configuration for the SSH Version 2 server is similar to the configuration for SSH version 1. Use the **ip ssh version** command to specify which version of SSH that you want to configure. If you do not configure this command, SSH by default runs in compatibility mode; that is, both SSH version 1 and SSH version 2 connections are honored.

**Note**

SSH version 1 is a protocol that has never been defined in a standard. If you do not want your device to fall back to the undefined protocol (version 1), you should use the **ip ssh version** command and specify version 2.

Use the **ip ssh rsa keypair-name** command to enable an SSH connection using Rivest, Shamir, and Adelman (RSA) keys that you have configured. If you configure the **ip ssh rsa keypair-name** command with a key-pair name, SSH is enabled if the key pair exists, or SSH will be enabled if the key pair is generated later. If you use this command to enable SSH, you do not need to configure a hostname and a domain name.

How to Configure NETCONF over SSHv2

Enabling SSH Version 2 Using a Hostname and Domain Name

Perform this task to configure your device for SSH version 2 using a hostname and domain name. You may also configure SSH version 2 by using the RSA key pair configuration (see [Enabling SSH Version 2 Using RSA Key Pairs](#), on page 68).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **hostname** *hostname*
4. **ip domain-name** *name*
5. **crypto key generate rsa**
6. **ip ssh** [*timeout seconds* | **authentication-retries** *integer*]
7. **ip ssh version 2**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	hostname <i>hostname</i> Example: Device(config)# hostname host1	Configures a hostname for your device.
Step 4	ip domain-name <i>name</i> Example: Device(config)# ip domain-name domain1.com	Configures a domain name for your device.
Step 5	crypto key generate rsa Example: Device(config)# crypto key generate rsa	Enables the SSH server for local and remote authentication.
Step 6	ip ssh [timeout <i>seconds</i> authentication-retries <i>integer</i>] Example: Device(config)# ip ssh timeout 120	(Optional) Configures SSH control variables on your device.
Step 7	ip ssh version 2 Example: Device(config)# ip ssh version 2	Specifies the version of SSH to be run on your device.

Enabling SSH Version 2 Using RSA Key Pairs

Perform this task to enable SSH version 2 without configuring a hostname or domain name. SSH version 2 will be enabled if the key pair that you configure already exists or if it is generated later. You may also configure SSH version 2 by using the hostname and domain name configuration. (See [“Enabling SSH Version 2 Using a Hostname and Domain Name, on page 67.”](#))

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip ssh rsa keypair-name** *keypair-name*
4. **crypto key generate rsa usage-keys label** *key-label* **modulus** *modulus-size*
5. **ip ssh** [*timeout seconds* | **authentication-retries** *integer*]
6. **ip ssh version 2**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip ssh rsa keypair-name <i>keypair-name</i> Example: Device(config)# ip ssh rsa keypair-name sshkeys	Specifies which RSA keypair to use for SSH usage. Note A Cisco device can have many RSA key pairs.
Step 4	crypto key generate rsa usage-keys label <i>key-label</i> modulus <i>modulus-size</i> Example: Device(config)# crypto key generate rsa usage-keys label sshkeys modulus 768	Enables the SSH server for local and remote authentication on the device. For SSH version 2, the modulus size must be at least 768 bits. Note To delete the RSA key pair, use the crypto key zeroize rsa command. After you have deleted the RSA command, you automatically disable the SSH server.
Step 5	ip ssh [<i>timeout seconds</i> authentication-retries <i>integer</i>] Example: Device(config)# ip ssh timeout 120	Configures SSH control variables on your device.
Step 6	ip ssh version 2 Example: Device(config)# ip ssh version 2	Specifies the version of SSH to be run on a device.

Starting an Encrypted Session with a Remote Device

Perform this task to start an encrypted session with a remote networking device. (You do not have to enable your device. SSH can be run in disabled mode.)

From any UNIX or UNIX-like device, the following command is typically used to form an SSH session:

```
ssh -2 -s user@router.example.com netconf
```

SUMMARY STEPS

1. Do one of the following:

- `ssh [-v {1 | 2}] [-c {3des| aes128-cbc | aes192-cbc| aes256-cbc}] [-m {hmac-md5 | hmac-md5-96 | hmac-sha1 | hmac-sha1-96}] [1 userid] [-o numberofpasswordprompts n] [-p port-num] {ip-addr | hostname} [command]`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>Do one of the following:</p> <ul style="list-style-type: none"> • <code>ssh [-v {1 2}] [-c {3des aes128-cbc aes192-cbc aes256-cbc}] [-m {hmac-md5 hmac-md5-96 hmac-sha1 hmac-sha1-96}] [1 <i>userid</i>] [-o <i>numberofpasswordprompts n</i>] [-p <i>port-num</i>] {<i>ip-addr</i> <i>hostname</i>} [<i>command</i>]</code> <p>Example:</p> <pre>Device# ssh -v 2 -c aes256-cbc -m hmac-sha1-96 -l user2 10.76.82.24</pre> <p>Example:</p> <pre>Device# ssh -v 2 -c aes256-cbc -m hmac-sha1-96 user2@10.76.82.24</pre>	<p>Starts an encrypted session with a remote networking device.</p> <p>The first example adheres to the SSH version 2 conventions. A more natural and common way to start a session is by linking the username with the hostname. For example, the second configuration example provides an end result that is identical to that of the first example.</p>

Troubleshooting Tips

The `ip ssh version` command can be used for troubleshooting your SSH configuration. By changing versions, you can determine which SSH version has a problem.

What to Do Next

For more information about the `ssh` command, see the Cisco IOS Security Command Reference.

Verifying the Status of the Secure Shell Connection

Perform this task to display the status of the SSH connection on your device.


Note

You can use the following **show** commands in user EXEC or privileged EXEC mode.

SUMMARY STEPS

1. `enable`
2. `show ssh`
3. `show ip ssh`

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	(Optional) Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	show ssh Example: Device# show ssh	Displays the status of SSH server connections.
Step 3	show ip ssh Example: Device# show ip ssh	Displays the version and configuration data for SSH.

Examples

The following output from the `show ssh` command displays status about SSH version 2 connections.

```
Device# show ssh
Connection Version Mode Encryption Hmac State
Username
1 2.0 IN aes128-cbc hmac-md5 Session started lab
```

```
1          2.0      OUT aes128-cbc hmac-md5      Session started      lab
%No SSHv1 server connections running.
```

The following output from the **show ip ssh** command displays the version of SSH that is enabled, the authentication timeout values, and the number of authentication retries.

```
Device# show ip ssh
SSH Enabled - version 2.0
Authentication timeout: 120 secs; Authentication retries: 3
```

Enabling NETCONF over SSHv2

Perform this task to enable NETCONF over SSHv2.

Before You Begin

SSHv2 must be enabled.



Note

There must be at least as many vty lines configured as there are concurrent NETCONF sessions.



Note

- A minimum of four concurrent NETCONF sessions must be configured.
- A maximum of 16 concurrent NETCONF sessions can be configured.
- NETCONF does not support SSHv1.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **netconf ssh** [*acl access-list-number*]
4. **netconf lock-time** *seconds*
5. **netconf max-sessions** *session*
6. **netconf max-message** *size*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	netconf ssh [acl access-list-number] Example: Device(config)# netconf ssh acl 1	Enables NETCONF over SSHv2. <ul style="list-style-type: none"> • Optionally, you can configure an access control list for this NETCONF session.
Step 4	netconf lock-time seconds Example: Device(config)# netconf lock-time 60	(Optional) Specifies the maximum time, in seconds, a NETCONF configuration lock is in place without an intermediate operation. <ul style="list-style-type: none"> • The valid range is 1 to 300. The default value is 10 seconds.
Step 5	netconf max-sessions session Example: Device(config)# netconf max-sessions 5	(Optional) Specifies the maximum number of concurrent NETCONF sessions allowed. <ul style="list-style-type: none"> • The valid range is 4 to 16. The default value is 4.
Step 6	netconf max-message size Example: Device(config)# netconf max-message 37283	(Optional) Specifies the maximum size, in kilobytes (KB), for the messages received in a NETCONF session. <ul style="list-style-type: none"> • The valid range is 1 to 2147483. The default value is infinite. • To set the maximum size to infinite, use the no netconf max-message command.

Configuration Examples for NETCONF over SSHv2

Example: Enabling SSHv2 Using a Hostname and Domain Name

```
configure terminal
hostname host1
ip domain-name example.com
crypto key generate rsa
ip ssh timeout 120
ip ssh version 2
```

Enabling Secure Shell Version 2 Using RSA Keys Example

The following example shows how to configure SSHv2 using RSA keys:

```
Device# configure terminal
Device(config)# ip ssh rsa keypair-name sshkeys
Device(config)# crypto key generate rsa usage-keys label sshkeys modulus 768
Device(config)# ip ssh timeout 120
Device(config)# ip ssh version 2
```

Starting an Encrypted Session with a Remote Device Example

The following example shows how to start an encrypted SSH session with a remote networking device, from any UNIX or UNIX-like device:

```
Device(config)# ssh -2 -s user@router.example.com netconf
```

Configuring NETCONF over SSHv2 Example

The following example shows how to configure NETCONF over SSHv2:

```
Device# configure terminal
Device(config)# netconf ssh acl 1
Device(config)# netconf lock-time 60
Device(config)# netconf max-sessions 5
Device(config)# netconf max-message 2345
Device# ssh-2 -s username@10.1.1.1 netconf
```

The following example shows how to get the configuration for loopback interface 113.

SUMMARY STEPS

1. First, send the “hello”:
2. Next, send the get-config request:

DETAILED STEPS

	Command or Action	Purpose
Step 1	<p>First, send the “hello”:</p> <p>Example:</p> <pre><?xml version="1.0" encoding="UTF-8"?> <hello><capabilities> <capability>urn:ietf:params:netconf:base:1.0</capability> <capability>urn:ietf:params:netconf:capability:writeable-running:1.0</capability> <capability>urn:ietf:params:netconf:capability:rollback-on-error:1.0</capability> <capability>urn:ietf:params:netconf:capability:startup:1.0</capability> <capability>urn:ietf:params:netconf:capability:url:1.0</capability> <capability>urn:cisco:params:netconf:capability:pi-data-model:1.0</capability></pre>	

	Command or Action	Purpose
	<pre><capability>urn:cisco:params:netconf:capabili?ty:notificati?on:1.0</capability> </capabilities> </hello>]]>]]></pre>	
Step 2	<p>Next, send the get-config request:</p> <p>Example:</p> <pre><?xml version="1.0"?> <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"xmlns:cpi="http://www.cisco.com/cpi_10/schema" message-id="101"> <get-config> <source> <running/> </source> <filter> <config-format-text-cmd> <text-filter-spec> interface Loopback113 </text-filter-spec> </config-format-text-cmd> </filter> </get-config> </rpc>]]>]]></pre>	

The following output is shown on the device:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="101"xmlns="\urn:ietf:params:netconf:base:1.0\">
  <data>
    <cli-config-data>
      interface Loopback113
      description test456
      no ip address
      load-interval 30
      end
    </cli-config-data>
  </data>
</rpc-rep?ly>]]>]]>
```

Additional References for NETCONF over SSHv2

Related Documents

Related Topic	Document Title
Cisco IOS Commands	Cisco IOS Master Command List, All Releases
NETCONF commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	<i>Cisco IOS Cisco Networking Services Command Reference</i>

Related Topic	Document Title
<p>IP access lists commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples</p> <p>Security commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples</p>	<i>Cisco IOS Security Command Reference</i>
IP access lists	IP Access List Overview and Creating an IP Access List and Applying It to an Interface modules in the Cisco IOS Security Configuration Guide: Securing the Data Plane.
Secure Shell and Secure Shell Version 2	“Configuring Secure Shell” module in the Cisco IOS Security Configuration Guide: Securing User Services.

Standards and RFCs

RFC	Title
RFC 2246	<i>The TLS Protocol Version 1.0</i>
RFC 4251	<i>The Secure Shell (SSH) Protocol Architecture</i>
RFC 4252	<i>The Secure Shell (SSH) Authentication Protocol</i>
RFC 4741	NETCONF Configuration Protocol
RFC 4742	Using the NETCONF Configuration Protocol over Secure Shell (SSH)

Technical Assistance

Description	Link
<p>The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/cisco/web/support/index.html

Feature Information for NETCONF over SSHv2

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 8: Feature Information for NETCONF over SSHv2

Feature Name	Releases	Feature Information
NETCONF over SSHv2	Cisco IOS XE Release 2.1 12.2(33)SB 12.2(33)SRA 12.2(33)SXI 12.4(9)T	The NETCONF over SSHv2 feature enables you to perform network configurations via the Cisco command-line interface (CLI) over an encrypted transport. The following commands were introduced or modified by this feature: netconf lock-time , netconf max-message , netconf max-sessions netconf ssh .



NETCONF Access for Configurations over BEEP

You can use the Network Configuration Protocol (NETCONF) over Blocks Extensible Exchange Protocol (BEEP) feature to send notifications of any configuration change over NETCONF. A notification is an event indicating that a configuration change has happened. The change can be a new configuration, deleted configuration, or changed configuration. The notifications are sent at the end of a successful configuration operation as one message showing the set of changes, rather than individual messages for each line in the configuration that is changed.

BEEP can use the Simple Authentication and Security Layer (SASL) profile to provide simple and direct mapping to the existing security model. Alternatively, NETCONF over BEEP can use the transport layer security (TLS) to provide a strong encryption mechanism with either server authentication or server and client-side authentication.

- [Finding Feature Information, page 79](#)
- [Prerequisites for NETCONF Access for Configurations over BEEP, page 80](#)
- [Restrictions for NETCONF Access for Configurations over BEEP, page 80](#)
- [Information About NETCONF Access for Configurations over BEEP, page 80](#)
- [How to Configure NETCONF Access for Configurations over BEEP, page 81](#)
- [Configuration Examples for NETCONF Access for Configurations over BEEP, page 86](#)
- [Additional References for NETCONF Access for Configurations over BEEP, page 86](#)
- [Feature Information for NETCONF Access for Configurations over BEEP, page 87](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for NETCONF Access for Configurations over BEEP

NETCONF over BEEP listeners require Simple Authentication and Security layer (SASL) to be configured.

Restrictions for NETCONF Access for Configurations over BEEP

You must be running a crypto image in order to configure BEEP using transport layer security (TLS).

Information About NETCONF Access for Configurations over BEEP

NETCONF over BEEP Overview

The NETCONF Access for Configurations over BEEP feature allows you to enable BEEP as the transport protocol to use during NETCONF sessions. Using NETCONF over BEEP, you can configure either the NETCONF server or the NETCONF client to initiate a connection, thus supporting large networks of intermittently connected devices, and those devices that must reverse the management connection where there are firewalls and Network Address Translators (NATs).

BEEP is a generic application protocol framework for connection-oriented, asynchronous interactions. It is intended to provide the features that traditionally have been duplicated in various protocol implementations. BEEP typically runs on top of Transmission Control Protocol (TCP) and allows the exchange of messages. Unlike HTTP and similar protocols, either end of the connection can send a message at any time. BEEP also includes facilities for encryption and authentication and is highly extensible.

The BEEP protocol contains a framing mechanism that permits simultaneous and independent exchanges of messages between peers. These messages are usually structured using XML. All exchanges occur in the context of a binding to a well-defined aspect of the application, such as transport security, user authentication, or data exchange. This binding forms a channel; each channel has an associated profile that defines the syntax and semantics of the messages exchanged.

The BEEP session is mapped onto the NETCONF service. When a session is established, each BEEP peer advertises the profiles it supports. During the creation of a channel, the client (the BEEP initiator) supplies one or more proposed profiles for that channel. If the server (the BEEP listener) creates the channel, it selects one of the profiles and sends it in a reply. The server may also indicate that none of the profiles are acceptable, and decline creation of the channel.

BEEP allows multiple data exchange channels to be simultaneously in use.

Although BEEP is a peer-to-peer protocol, each peer is labeled according to the role it is performing at a given time. When a BEEP session is established, the peer that awaits new connections is the BEEP listener. The other peer, which establishes a connection to the listener, is the BEEP initiator. The BEEP peer that starts an exchange is the client, and the other BEEP peer is the server. Typically, a BEEP peer that acts in the server role also performs in the listening role. However, because BEEP is a peer-to-peer protocol, the BEEP peer that acts in the server role is not required to also perform in the listening role.

NETCONF over BEEP and SASL

The SASL is an Internet standard method for adding authentication support to connection-based protocols. SASL can be used between a security appliance and an Lightweight Directory Access Protocol (LDAP) server to secure user authentication.

BEEP listeners require SASL to be configured.

NETCONF over BEEP and TLS

The TLS is an application-level protocol that provides for secure communication between a client and server by allowing mutual authentication, the use of hash for integrity, and encryption for privacy. TLS relies upon certificates, public keys, and private keys.

Certificates are similar to digital ID cards. They prove the identity of the server to clients. Each certificate includes the name of the authority that issued it, the name of the entity to which the certificate was issued, the entity's public key, and time stamps that indicate the certificate's expiration date.

Public and private keys are the ciphers used to encrypt and decrypt information. Although the public key is shared, the private key is never given out. Each public-private key pair works together. Data encrypted with the public key can be decrypted only with the private key.

NETCONF over BEEP and Access Lists

You can optionally configure access lists for use with NETCONF over SSHv2 sessions. An access list is a sequential collection of permit and deny conditions that apply to IP addresses. The Cisco software tests addresses against the conditions in an access list one by one. The first match determines whether the software accepts or rejects the address. Because the software stops testing conditions after the first match, the order of the conditions is critical. If no conditions match, the software rejects the address.

The two main tasks involved in using access lists are as follows:

- 1 Creating an access list by specifying an access list number or name and access conditions.
- 2 Applying the access list to interfaces or terminal lines.

For more information about configuring access lists, see “IP Access List Overview” and “Creating an IP Access List and Applying It to an Interface” modules in *Security Configuration Guide: Securing the Data Plane*.

How to Configure NETCONF Access for Configurations over BEEP

Configuring an SASL Profile

To enable NETCONF over BEEP using SASL, you must first configure an SASL profile, which specifies which users are allowed access into the device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sasl profile** *profile-name*
4. **mechanism di gest-md5**
5. **server** *user-name* **password** *password*
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	sasl profile <i>profile-name</i> Example: Device(config)# sasl profile beep	Configures an SASL profile and enters SASL profile configuration mode.
Step 4	mechanism di gest-md5 Example: Device(config-SASL-profile)# mechanism digest-md5	Configures the SASL profile mechanism.
Step 5	server <i>user-name</i> password <i>password</i> Example: Device(config-SASL-profile)# server user1 password password1	Configures an SASL server.
Step 6	exit Example: Device(config)# exit	Exits global configuration mode and returns to privileged EXEC mode.

Enabling NETCONF over BEEP

Before You Begin

- There must be at least as many vty lines configured as there are concurrent NETCONF sessions.
- If you configure NETCONF over BEEP using SASL, you must first configure an SASL profile.



Note

- A minimum of four concurrent NETCONF sessions must be configured.
- A maximum of 16 concurrent NETCONF sessions can be configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto key generate rsa general-keys**
4. **crypto pki trustpoint** *name*
5. **enrollment url** *url*
6. **subject-name** *name*
7. **revocation-check** *method1* [*method2* [*method3*]]
8. **exit**
9. **crypto pki authenticate** *name*
10. **crypto pki enroll** *name*
11. **netconf lock-time** *seconds*
12. **line vty** *line-number* [*ending-line-number*]
13. **netconf max-sessions** *session*
14. **netconf beep initiator** {*hostname* | *ip-address*} *port-number* **user** *sasl-user* **password** *sasl-password* [**encrypt** *trustpoint*] [**reconnect-time** *seconds*]
15. **netconf beep listener** [*port-number*] [**acl** *access-list-number*] [**sasl** *sasl-profile*] [**encrypt** *trustpoint*]
16. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	crypto key generate rsa general-keys Example: Device(config)# crypto key generate rsa general-keys	Generates Rivest, Shamir, and Adelman (RSA) key pairs and specifies that the general-purpose key pair should be generated. Perform this step only once.
Step 4	crypto pki trustpoint name Example: Device(config)# crypto pki trustpoint my_trustpoint	Declares the trustpoint that your router should use and enters ca-trustpoint configuration mode.
Step 5	enrollment url url Example: Device(ca-trustpoint)# enrollment url http://10.2.3.3:80	Specifies the enrollment parameters of a certification authority (CA).
Step 6	subject-name name Example: Device(ca-trustpoint)# subject-name CN=dns_name_of_host.com	Specifies the subject name in the certificate request. Note The subject name should be the Domain Name System (DNS) name of the device.
Step 7	revocation-check method1 [method2 [method3]] Example: Device(ca-trustpoint)# revocation-check none	Checks the revocation status of a certificate.
Step 8	exit Example: Device(ca-trustpoint)# exit	Exits ca-trustpoint configuration mode and returns to global configuration mode.
Step 9	crypto pki authenticate name Example: Device(config)# crypto pki authenticate my_trustpoint	Authenticates the certification authority (by getting the certificate of the CA).

	Command or Action	Purpose
Step 10	crypto pki enroll <i>name</i> Example: <pre>Device(config)# crypto pki enroll my_trustpoint</pre>	Obtains the certificate or certificates for your router from CA.
Step 11	netconf lock-time <i>seconds</i> Example: <pre>Device(config)# netconf lock-time 60</pre>	(Optional) Specifies the maximum time a NETCONF configuration lock is in place without an intermediate operation. The valid value range for the seconds argument is 1 to 300 seconds. The default value is 10 seconds.
Step 12	line vty <i>line-number</i> [<i>ending-line-number</i>] Example: <pre>Device(config)# line vty 0 15</pre>	Identifies a specific virtual terminal line for remote console access. You must configure the same number of vty lines as maximum NETCONF sessions.
Step 13	netconf max-sessions <i>session</i> Example: <pre>Device(config)# netconf max-sessions 16</pre>	(Optional) Specifies the maximum number of concurrent NETCONF sessions allowed.
Step 14	netconf beep initiator <i>{hostname ip-address}</i> <i>port-number</i> user <i>sasl-user</i> password <i>sasl-password</i> [encrypt <i>trustpoint</i>] [reconnect-time <i>seconds</i>] Example: <pre>Device(config)# netconf beep initiator host1 23 user user1 password password1 encrypt 23 reconnect-time 60</pre>	(Optional) Specifies BEEP as the transport protocol for NETCONF sessions and configures a peer as the BEEP initiator. Note Perform this step to configure a NETCONF BEEP initiator session. You can also optionally configure a BEEP listener session.
Step 15	netconf beep listener [<i>port-number</i>] [acl <i>access-list-number</i>] [sasl <i>sasl-profile</i>] [encrypt <i>trustpoint</i>] Example: <pre>Device(config)# netconf beep listener 26 acl 101 sasl profile1 encrypt 25</pre>	(Optional) Specifies BEEP as the transport protocol for NETCONF and configures a peer as the BEEP listener. Note Perform this step to configure a NETCONF BEEP listener session. You can also optionally configure a BEEP initiator session.
Step 16	exit Example: <pre>Device(config)# exit</pre>	Exits global configuration mode and returns to privileged EXEC mode.

Configuration Examples for NETCONF Access for Configurations over BEEP

Example: Enabling NETCONF over BEEP

```

Device# configure terminal
Device(config)# crypto key generate rsa general-keys

Device(ca-trustpoint)# crypto pki trustpoint my_trustpoint

Device(ca-trustpoint)# enrollment url http://10.2.3.3:80
Device(ca-trustpoint)# subject-name CN=dns_name_of_host.com
Device(ca-trustpoint)# revocation-check none
Device(ca-trustpoint)# crypto pki authenticate my_trustpoint

Device(ca-trustpoint)# crypto pki enroll my_trustpoint

Device(ca-trustpoint)# line vty 0 15

Device(ca-trustpoint)# exit
Device(config)# netconf lock-time 60

Device(config)# netconf max-sessions 16

Device(config)# netconf beep initiator host1 23 user my_user password my_password encrypt
my_trustpoint reconnect-time 60

Device(config)# netconf beep listener 23 sasl user1 encrypt my_trustpoint

```

Additional References for NETCONF Access for Configurations over BEEP

Related Documents

Related Topic	Document Title
Cisco IOS Commands	Cisco IOS Master Commands List, All Releases
NETCONF commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	<i>Cisco IOS Cisco Networking Services Command Reference</i>

Standards and RFCs

Standard/RFC	Title
RFC 2222	<i>Simple Authentication and Security Layer (SASL)</i>
RFC 3080	<i>The Blocks Extensible Exchange Protocol Core</i>

Standard/RFC	Title
RFC 4741	<i>NETCONF Configuration Protocol</i>
RFC 4744	<i>Using the NETCONF Protocol over the Blocks Extensible Exchange Protocol (BEEP)</i>

Technical Assistance

Description	Link
<p>The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/cisco/web/support/index.html

Feature Information for NETCONF Access for Configurations over BEEP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 9: Feature Information for NETCONF Access for Configurations over BEEP

Feature Name	Releases	Feature Information
NETCONF Access for Configurations over BEEP	Cisco IOS XE Release 2.1 12.2(33)SB 12.2(33)SRB 12.2(33)SXI 12.4(9)T	<p>The NETCONF over BEEP feature allows you to enable either the NETCONF server or the NETCONF client to initiate a connection, thus supporting large networks of intermittently connected devices and those devices that must reverse the management connection where there are firewalls and network address translators (NATs).</p> <p>The following commands were introduced or modified by this feature: netconf beep initiator, netconf beep listener.</p>