

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.

- Prerequisites for Cisco Networking Services, on page 1
- Restrictions for Cisco Networking Services, on page 2
- Information About Cisco Networking Services, on page 2
- How to Configure Cisco Networking Services, on page 8
- Configuration Examples for Cisco Networking Services, on page 15
- Additional References, on page 19
- Feature Information for Cisco Networking Services, on page 20

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 **cns 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

Procedure

	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** *name* **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** *config-text* **Example:** `Device(config-templ-conn)# cli encapsulation ppp`	Specifies commands to configure the interface.
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** *name* [**retry-interval** *interval-seconds*] [**retries** *number-retries*] [**timeout** *timeout-seconds*] [**sleep** *sleep-seconds*]	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.

	Command or Action	Purpose
	Example: `Device(config)# cns connect profile-1 retry-interval 15 timeout 90`	

Step 8	Do one of the following:
	• **discover** {**line** *line-type* | **controller** *controller-type* | **interface** [*interface-type*]}
	• **template** *name*
	Example:
	`Device(config-cns-conn)# discover interface serial`
	Example:
	`Device(config-cns-conn)# template template-1`
		(Optional) Configures a generic bootstrap configuration.
		• **discover** —Defines the interface parameters within a Cisco Networking Services connect profile for connecting to the Cisco Networking Services configuration engine.
		or
		• **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** {*host-name* | *ip-address*} [**encrypt**] [*port-number*] [**page** *page*] [**syntax-check**] [**no-persist**] [**source** *interface name*] [**status** *url*] [**event**] [**inventory**] **Example:** `Device(config)# cns config initial 10.1.1.1 no-persist`	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. **Note** The optional **encrypt** keyword is available only in images that support Secure Socket Layer (SSL). **Caution** If you write the new configuration to NVRAM by omitting the **no-persist**keyword, the original bootstrap configuration is overwritten.
Step 11	**exit** **Example:** `Device(config)# exit`	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

Procedure

	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 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. • 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. • Use the **snmp** keyword to specify that nongranular encapsulation is used when Cisco Networking Services notifications are sent.

	Command or Action	Purpose
		• 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. • 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. • 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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables higher privilege levels, such as privileged EXEC mode. • 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. • This command displays a dump of XML that would be sent out in response to an image agent inventory request message. The 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 image**privileged 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 template1
 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	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. 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**.