Cisco Networking Services

Last Updated: December 20, 2011

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

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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 CNS

- Configure the remote router to support the CNS configuration agent and the CNS event agent.
- Configure a transport protocol on the remote router that is compatible with the remote router’s external interface. The following table lists the supported transport protocols that can be used depending on the router interface.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Table 1 Router Interface and Transport Protocols Required by CNS Services

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

Restrictions for CNS

CNS Configuration Engine

- The CNS 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 CNS configuration engine before installation of the remote router.
- The user of CNS Flow-Through Provisioning and the CNS configuration engine must be familiar with designing network topologies, designing configuration templates, and using the CNS configuration engine.

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.

Remote Router

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

Information About CNS

CNS

CNS 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, CNS has been designed to provide “plug-and-play” network services using a central directory service and distributed agents. CNS features include CNS configuration and event agents and a Flow-Through Provisioning structure. The configuration and event agents use a CNS configuration engine to provide methods for automating initial Cisco IOS XE 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 CNS Flow-Through Provisioning uses the CNS configuration and event agents to provide an automated workflow, eliminating the need for an on-site technician.

CNS Configuration Agent

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

Initial CNS Configuration

When a routing device first comes up, it connects to the configuration server component of the CNS 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 CNS 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 CNS 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 CNS web server for transmission to the routing device.

The CNS configuration agent accepts the configuration file from the CNS 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 CNS configuration engine to automatically install the initial CNS configuration, see the *Cisco CNS Configuration Engine Administrator’s Guide* at http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/cns/ce/rel13/ag13/index.htm .

Incremental CNS Configuration

Once the network is up and running, new services can be added using the CNS 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 CNS Configuration Agent feature allows the device configuration to be synchronized with other dependent network activities.

CNS Config Retrieve Enhancement with Retry and Interval

The Cisco Networking Services (CNS) 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.

CNS Interactive CLI

The CNS Interactive CLI feature provides a XML interface that allows you to send interactive commands to a router, 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.

CNS IDs

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

The network administrator must ensure a match between the CNS agent ID as defined on the routing device and the CNS 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 CNS agent ID is always set to the hostname. If the hostname changes, the CNS agent ID also changes. If the CNS 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 CNS agent ID does not address security issues.

CNS Password

The CNS password is used to authenticate the CNS device. You must configure the CNS password the first time a router is deployed, and the CNS password must be the same as the bootstrap password set on the Configuration Engine (CE). If both the router and the CE bootstrap password use their default settings, a newly deployed router will be able to connect to the CE.

Once connected, the CE manages the CNS password. Network administrators must ensure not to change the CNS password. If the CNS password is changed, connectivity to the CE will be lost.

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 CNS commands, Command Scheduler now has a broader application. Using the CNS image agent feature, remote routers 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 router.

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.

CNS Zero Touch

The CNS Zero Touch feature provides a zero touch deployment solution where the router contacts a CNS 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 CNS 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 CNS connect functionality is configured with a set of CNS connect templates. A CNS connect profile is created for connecting to the CNS configuration engine and to implement the CNS connect templates on a Customer Premise Equipment (CPE) router. CNS connect variables can be used as placeholders within a CNS connect template configuration. These variables, such as the active DLCI, are substituted with real values before the CNS connect templates are sent to the router’s parser.

To use the zero touch functionality, the router that is to be initialized must have a generic bootstrap configuration. This configuration includes CNS connect templates, CNS connect profiles, and the **cns config initial** command. This command initiates the CNS connect function.

The CNS connect functionality performs multiple ping iterations through the router’s interfaces and lines, as well as any available controllers. For each iteration, the CNS connect function attempts to ping the CNS configuration engine. If the ping is successful, the pertinent configuration information can be downloaded from the CNS configuration engine. If connectivity to the CNS configuration engine is unsuccessful, the

CNS connect function removes the configuration applied to the selected interface, and the CNS connect process restarts with the next available interface specified by the CNS connect profile.

The CNS Zero Touch feature provides the following benefits:

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

How to Configure CNS

- Deploying the CNS Router, page 6
- Configuring CNS Security Features, page 9
- Configuring Command Scheduler Policy Lists and Occurrences, page 11
- Configuring Advanced CNS Features, page 14

Deploying the CNS Router

Perform this task to manually install an initial CNS configuration.

Your remote router arrives from the factory with a bootstrap configuration. Upon initial power-on, the router automatically pulls a full initial configuration from the CNS 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 CNS Configuration, page 6
- Incremental Configuration, page 6

Initial CNS Configuration

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

CNS assigns the remote router 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 router and invokes the CNS agent to download the initial configuration from the CNS configuration engine.

Incremental Configuration

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

Before you can configure an incremental configuration, CNS must be operational and the required CNS 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:** `Router enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cns template connect** *name* **Example:** `Router(config)# cns template connect template-1`	Enters CNS template connect configuration mode and defines the name of a CNS connect template.

	Command or Action	Purpose
Step 4	**cli** *config-text* **Example:** `Router(config-templ-conn)# cli encapsulation ppp`	Specifies commands to configure the interface.
Step 5	Repeat Step 4 to add all required CLI commands. **Example:** `Router(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:** `Router(config-templ-conn)# exit`	Exits CNS template connect configuration mode and completes the configuration of a CNS 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*] **Example:** `Router(config)# cns connect profile-1 retry-interval 15 timeout 90`	Enters CNS connect configuration mode and defines the parameters of a CNS connect profile for connecting to the CNS configuration engine.

	Command or Action	Purpose

Step 8	Do one of the following:
	• **discover** {**line** *line-type* | **controller** *controller-type* | **interface** [*interface-type*]}
	•
	• **template** *name*
	Example:
	`Router(config-cns-conn)# discover interface serial`
	Example:
	Example:
	`Router(config-cns-conn)# template template-1`
		(Optional) Configures a generic bootstrap configuration.
		• **discover** --Defines the interface parameters within a CNS connect profile for connecting to the CNS configuration engine.
		or
		• **template** --Specifies a list of CNS connect templates within a CNS connect profile to be applied to a router's configuration.

Step 9	**exit** **Example:** `Router(config-cns-conn)# exit`	Exits CNS 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:** `Router(config)# cns config initial 10.1.1.1 no-persist`	Starts the CNS configuration agent, connects to the CNS 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:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring CNS Security Features

Perform this task to configure CNS security features.

- CNS Trusted Servers, page 10

CNS Trusted Servers

Use the **cns trusted-server** command to specify a trusted server for an individual CNS agent or for all the CNS agents. To avoid security violations, you can build a list of trusted servers from which CNS agents can receive messages. An attempt to connect to a server not on the list will result in an error message being displayed.

Configure a CNS trusted server when a CNS agent will redirect its response to a server address that is not explicitly configured on the command line for the specific CNS agent. For example, the CNS exec agent may have one server configured but receive a message from the CNS event bus that overrides the configured server. The new server address has not been explicitly configured, so the new server address is not a trusted server. An error will be generated when the CNS exec agent tries to respond to this new server address unless the **cns trusted-server** command has been configured for the new server address.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cns trusted-server** {**all-agents** | **config** | **event** | **exec** | **image**} *name*
4. **cns message format notification** [**version 1**| **version 2**]
5. **cns aaa authentication** *authentication-method*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cns trusted-server** {**all-agents** | **config** | **event** | **exec** | **image**} *name* **Example:** `Router(config)# cns trusted-server event 10.19.2.5`	Configures a CNS trusted server for the specified hostname or IP address.

	Command or Action	Purpose
Step 4	**cns message format notification** [**version 1**| **version 2**] **Example:** `Router(config)# cns message format notification version 1`	Configures the message format for notification messages from a CNS device. Received messages which do not conform to the configured message format are rejected. Use version 1 to configure the non-SOAP message format. Use version 2 for SOAP message format.
Step 5	**cns aaa authentication** *authentication-method* **Example:** `Router(config)# cns aaa authentication method1`	Enables CNS AAA options. **Note** The authentication methods must be configured within AAA.

Configuring Command Scheduler Policy Lists and Occurrences

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

- Command Scheduler Policy Lists, page 11
- Command Scheduler Occurrences, page 11
- Examples, page 14
- Troubleshooting Tips, page 14

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

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.

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* {**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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**kron policy-list** *list-name* [**conditional**] **Example:** `Router(config)# kron policy-list cns-weekly`	Specifies a name for a new or existing Command Scheduler policy list and enters kron-policy configuration mode. • If the *list-name* is new, a new policy list structure is created. • If the *list-name* 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	**cli** *command* **Example:** `Router(config-kron-policy)# cli cns image retrieve server https://10.19.2.3/cnsweek/ status https://10.19.2.3/cnsstatus/week/`	Specifies the fully-qualified EXEC command and associated syntax to be added as an entry in the specified Command Scheduler policy list. • 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. **Note** EXEC commands that generate a prompt or can be terminated using keystrokes will cause an error.
Step 5	**exit** **Example:** `Router(config-kron-policy)# exit`	Exits kron-policy configuration mode and returns the router to global configuration mode.

Step 6	**kron occurrence** *occurrence-name* {**in**[[*numdays*:]*numhours*:]*nummin* | **at** *hours*:*min*[[*month*] *day-of-month*] [*day-of-week*]} {**oneshot** | **recurring**| **system-startup**}
	Example:
	`Router(config)# kron occurrence may user sales at 6:30 may 20 oneshot`
		Specifies a name and schedule for a new or existing Command Scheduler occurrence and enters kron-occurrence configuration mode.
		• 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	**policy-list** *list-name* **Example:** `Router(config-kron-occurrence)# policy-list sales-may`	Specifies a Command Scheduler policy list. • Each entry is added to the occurrence list in the order in which it is configured. **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.

	Command or Action	Purpose
Step 8	**exit** **Example:** `Router(config-kron-occurrence)# exit`	Exits kron-occurrence configuration mode and returns the router to global configuration mode. • Repeat this step to exit global configuration mode.
Step 9	**show kron schedule** **Example:** `Router# 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:

```
Router# 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 router operations.

Configuring Advanced CNS Features

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

Some other advanced features allow you to use the Software Developer's Toolkit (SDK) to specify how CNS 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	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	**cns mib-access encapsulation** {**snmp** | **xml**[**size** *bytes*]}
	Example:
	`Router(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:
	`Router(config)# cns notifications encapsulation xml`
		(Optional) Specifies the type of encapsulation to use when sending CNS notifications.
		• Use the **snmp** keyword to specify that nongranular encapsulation is used when CNS notifications are sent.
		• Use the **xml** keyword to specify that granular encapsulation is used when CNS notifications are sent.

Step 5	**cns inventory** **Example:** `Router(config)# cns inventory`	Enables the CNS inventory agent and enters CNS inventory mode. • An inventory of the router’s line cards and modules is sent to the CNS configuration engine.
Step 6	**transport event** **Example:** `Router(cns-inv)# transport event`	Specifies that inventory requests are sent out with each CNS inventory agent message.

	Command or Action	Purpose
Step 7	**exit** **Example:** `Router(cns-inv)# exit`	Exits CNS inventory mode and returns to global configuration mode. • Repeat this command to return to privileged EXEC mode.

Configuration Examples for CNS

- Deploying the CNS Router Example, page 16
- Enabling and Configuring CNS Agents Example, page 16
- Command Scheduler Policy Lists and Occurrences Examples, page 17
- Retrieving a CNS Configuration from a Server Examples, page 17
- Using the CNS Zero Touch Solution Examples, page 18

Deploying the CNS Router Example

The following example shows an initial configuration on a remote router. The hostname of the remote router is the unique ID. The CNS 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
```

Enabling and Configuring CNS Agents Example

The following example shows various CNS 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 router. The CNS configuration engine IP address is 172.28.129.22, and the port number is 80. The CNS event agent is enabled with an IP address of 172.28.129.24. Until the CNS event agent is enabled, no other CNS agents are operational.

```
 cns config partial 172.28.129.22 80
 cns event 172.28.129.24 source 172.22.2.1
 exit
```

Command Scheduler Policy Lists and Occurrences Examples

In the following example, a Command Scheduler policy named cns-weekly is configured to run two sets of EXEC CLI involving CNS 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 CNS 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 CNS 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 CNS 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
```

Retrieving a CNS Configuration from a Server Examples

Retrieving Configuration Data from the CNS 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 CNS 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-Shft-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 CNS 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
```

Using the CNS Zero Touch Solution Examples

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 Router Interfaces

The following examples show how to configure a standard serial interface and a serial interface bound to a controller on an aggregator router (also known as the DCE). In order for connectivity to be established, the aggregator router 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 router:

```
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 router:

```
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

The following sections provide references related to the CNS feature.

Related Documents

Related Topic	Document Title
CNS commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Network Management Command Reference*
CNS Configuration Engine	*Cisco Intelligence Engine 2100 Configuration Registrar Manual , Release 1.1 or later* Cisco CNS Configuration Engine Administrator’s Guide

Standards

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

MIBs

MIB	MIBs Link
The CNS Flow-Through Provisioning feature provides two mechanisms for accessing MIBs: a nongranular mechanism using SNMP encapsulation and a granular mechanism using XML encapsulation. These mechanisms enable you to access the MIBS currently available in the remote router. The MIBS currently available depend on the router platform and Cisco IOS XE release.	To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

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

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for CNS

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 CNS

Feature Name	Releases	Feature Information
CNS	Cisco IOS XE Release 2.1	The CNS feature is a collection of services that can provide remote event-driven configuring of Cisco IOS XE networking devices and remote execution of some command-line interface (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**.

Feature Name	Releases	Feature Information
CNS Config Retrieve Enhancement with Retry and Interval	Cisco IOS XE Release 2.1	The Cisco Networking Services (CNS) 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. • CNS Config Retrieve Enhancement with Retry and Interval, page 4 • Retrieving a CNS Configuration from a Server, page 27 • Retrieving a CNS Configuration from a Server: Example, page 43 The following command was modified by this feature: **cns config retrieve**.
CNS Interactive CLI	Cisco IOS XE Release 2.1	The CNS Interactive CLI feature introduces a new XML interface that allows you to send interactive commands to a router, such as commands that generate prompts for user input.

Feature Name	Releases	Feature Information
CNS Zero Touch	Cisco IOS XE Release 2.1	The CNS Zero Touch feature provides a zero touch deployment solution where the router contacts a CNS configuration engine to retrieve its full configuration automatically. The following commands were introduced or modified by this feature: **cli (cns)**, **cns config connect-intf**, **cns connect**, **cns template connect**, **config-cli**, **discover (cns)**, **line-cli, template (cns)**. **Note** The **cns config connect-intf**command was replaced by the **cns connect**and **cns template connect** commands. **Note** The **config-cli** and **line-cli** commands were replaced by the **cli (cns)** command.
Command Scheduler	Cisco IOS XE Release 2.1	The Command Scheduler feature provides the ability to schedule some EXEC command-line interface (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
CNS Configuration Agent	Cisco IOS XE Release 2.1	The CNS Configuration Agent feature supports routing devices by providing the following: • Initial configurations • Incremental (partial) configurations • Synchronized configuration updates 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**.

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.