Cisco Networking Services

First Published: November 20, 2006
Last Updated: June 29, 2007

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 in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for CNS" section on page 51.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

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

Contents

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

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. Table 1 lists the supported transport protocols that can be used depending on the router interface.
- Create the configuration template in the CNS configuration-engine provisioning database. (This task is best done by a senior network designer.)

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

CNS Image Agent

- Determine where to store the Cisco IOS images on a file server to make the image available to many other networking devices. If the CNS Event Bus is to be used to store and distribute the images, the CNS event agent must be configured.
- Set up a file server to enable the networking devices to download the new images. Protocols such as TFTP, HTTP, HTTPS, and rcp can be used.
- Determine how to handle error messages generated by CNS image agent operations. Error messages can be sent to the CNS Event Bus or an HTTP or HTTPS URL.

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.

CNS Image Agent

During automated image loading operations you must try to prevent the Cisco IOS device from losing connectivity with the file server that is providing the image. Image reloading is subject to memory issues and connection issues. Boot options must also be configured to allow the Cisco IOS device to boot another image if the first image reload fails. For more details see the "Managing Configuration Files" module of the *Cisco IOS Configuration Fundamentals Configuration Guide*, Release 12.4.

CNS Frame Relay Zero Touch

The CNS Frame Relay Zero Touch solution does not support switched virtual circuits (SVCs).

The Frame Relay zero touch solution does not support IP over PPP over Frame Relay because routing to an interface (or subinterface) that supports IP over PPP over Frame Relay is not possible.

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 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

To configure CNS, you should understand the following concepts:

- CNS, page 4
- CNS Configuration Agent, page 4
- Initial CNS Configuration, page 4
- Incremental CNS Configuration, page 5
- Synchronized Configuration, page 5
- CNS Config Retrieve Enhancement with Retry and Interval, page 5
- CNS EXEC Agent, page 5
- CNS Event Agent, page 5
- CNS Image Agent, page 5
- CNS Results Messages, page 6
- CNS Message Formats, page 6
- CNS Security Enhancement, page 9
- CNS Interactive CLI, page 10
- CNS IDs, page 10
- Command Scheduler, page 10
- CNS Flow-Through Provisioning, page 11
- CNS Zero Touch, page 15
- CNS Frame Relay Zero Touch, page 15

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 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 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 EXEC Agent

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

CNS Event Agent

Although other CNS agents may be configured, no other CNS agents are operational until the **cns event** command is entered because the CNS event agent provides a transport connection to the CNS event bus for all other CNS agents. The other CNS agents use the connection to the CNS event bus to send and receive messages. The CNS event agent does not read or modify the messages.

CNS Image Agent

Administrators maintaining large networks of Cisco IOS devices need an automated mechanism to load image files onto large numbers of remote devices. Existing network management applications are useful to determine which images to run and how to manage images received from the Cisco online software center. Other image distribution solutions do not scale to cover thousands of devices and cannot distribute images to devices behind a firewall or using Network Address Translation (NAT). The CNS image agent enables the managed device to initiate a network connection and request an image download allowing devices using NAT, or behind firewalls, to access the image server.

The CNS image agent can be configured to use the CNS Event Bus. To use the CNS Event Bus, the CNS event agent must be enabled and connected to the CNS event gateway in the CNS Configuration Engine. The CNS image agent can also use an HTTP server that understands the CNS image agent protocol. Deployment of CNS image agent operations can use both the CNS Event Bus and an HTTP server.

CNS Results Messages

When a partial configuration has been received by the router, each line of the configuration will be applied in the same order as it was received. If the Cisco IOS 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 router, 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 CNS event bus after the partial configuration is complete. The CNS event bus will display one of the following status messages:

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

In Cisco IOS Releases 12.4(4)T, 12.2 (33)SRA, and later releases, 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.

CNS Message Formats

SOAP Message Format

Using the Service-Oriented Access Protocol (SOAP) protocol provides a way to format the layout of CNS 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 CNS notification messages to authenticate user credentials.

CNS 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 CNS request message to the Cisco IOS 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"
wsse:usernameToken SOAP:mustUnderstand="0"
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:AgentCNS_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 CNS agent mentioned in the Agent field.

Response Message

The following is the format of a CNS response message from the Cisco IOS 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 IOS 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 CNS notification message sent from the Cisco IOS 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 IOS 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 IOS 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
```

CNS Security Enhancement

Before the introduction of the CNS Security Enhancement feature, the CNS message format did not support security. Using the new CNS SOAP message structure, the username and password are authenticated.

If authentication, authorization, and accounting (AAA) is configured, then CNS SOAP messages will be authenticated with AAA. If AAA is not configured, there will be no authentication. For backward compatibility, CNS will support the existing non-SOAP message format and will respond accordingly without security.

The **cns aaa authentication** command is required to turn on CNS Security Enhancement. This command determines whether the CNS messages are using AAA security or not. If the **cns aaa authentication** command is configured, then all incoming SOAP messages into the device are authenticated by AAA.

CNS Interactive CLI

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. 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 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.

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 Flow-Through Provisioning

Cisco Networking Services (CNS) Flow-Through Provisioning provides the infrastructure for automated configuration of large numbers of network devices. Based on CNS event and configuration agents, it eliminates the need for an onsite technician to initialize the device. The result is an automated workflow from initial subscriber-order entry through Cisco manufacturing and shipping to final device provisioning and subscriber billing. This functionality focuses on a root problem of today's service-provider and other similar business models: use of human labor in activating service.

To achieve such automation, CNS Flow-Through Provisioning relies on standardized configuration templates that you create. However, the use of such templates requires a known fixed hardware configuration, uniform for all subscribers. There is no way to achieve this without manually prestaging each line card or module within each chassis. While the inventory within a chassis is known at time of manufacture, controlling which line cards or modules are in which slots thereafter is labor-intensive and error-prone.

To overcome these difficulties, CNS Flow-Through Provisioning defines a new set of Cisco IOS commands—the **cns** commands. When a remote router is first powered on, these commands do the following:

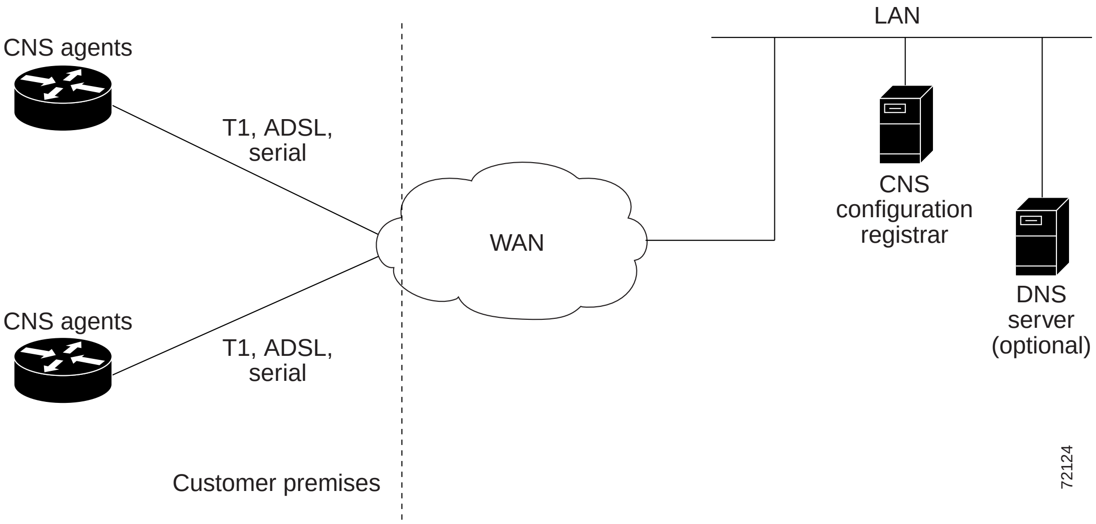
1. To each router interface in turn, applies a preset temporary bootstrap configuration that pings the CNS configuration engine, until a ping is successful and the connecting interface thus determined.
2. Connects, by way of software called a CNS agent, to a CNS configuration engine housed in a Cisco IE2100 device.
3. Passes to the CNS configuration engine a device-unique ID, along with a human-readable description of the router's line-card or module inventory by product number and location, in XML format.

In turn, the configuration engine does the following:

1. Locates in a Lightweight Directory Access Protocol (LDAP) directory, based on the device IDs, a predefined configuration template for the main chassis and subconfiguration template for each line card or module.
2. Substitutes actual slot numbers from the chassis inventory for the template's slot-number parameters, thus resolving the templates into subscriber-specific configurations that match the true line-card or module slot configuration.
3. Downloads this initial configuration to the target router. The CNS agent directly applies the configuration to the router.

Figure 1 shows the CNS Flow-Through Provisioning architecture.

Figure 1 CNS Flow-Through Provisioning Architecture



Configurations

CNS Flow-Through Provisioning involves three different types of configuration on the remote router:

- Bootstrap configuration

  You specify the preset bootstrap configuration on which this solution depends as part of your order from Cisco using Cisco Configuration Express, an existing service integrated with the Cisco.com order-entry tool. You specify a general-subscriber nonspecific bootstrap configuration that provides connectivity to the CNS configuration engine. Cisco then applies this configuration to all the devices of that order in a totally automated manufacturing step. This configuration runs automatically on power-on.

- Initial configuration

  The CNS configuration engine downloads an initial configuration, once only, to replace the temporary bootstrap configuration. You can either save or not save it in the router's nonvolatile NVRAM memory:

  – If you save the configuration, the bootstrap configuration is overwritten.

  – If you do not save the configuration, the download procedure repeats each time that the router powers off and then back on. Repeating the download procedure enables the router to update to the current Cisco IOS configuration without intervention.

- Incremental (partial) configuration

  On subsequent reboot, incremental or partial configurations are performed to update the configuration without the network having to shut down. Such configurations can be delivered either in a push operation that you initiate or a pull operation on request from the router.

Unique IDs

Key to this solution is the capability to associate, with each device, a simple, manageable, and unique ID that is compatible with your systems for order entry, billing, provisioning, and shipping and can also link your order-entry system to the Cisco order-fulfillment system. Such an ID must have the following characteristics:

- Be available from manufacturing as part of order fulfillment
- Be recordable on the shipping carton and chassis
- Be available to the device’s Cisco IOS software
- Be modifiable after the device is first powered up
- Be representative of both a specific chassis and a specific entry point into your network

To define such an ID, CNS Flow-Through Provisioning equips the CNS agent with a new set of commands—the **cns** commands—with which you specify how configurations should be done and, in particular, how the system defines unique IDs. You enable the Cisco IOS software to auto-discover the unique ID according to directions that you specify and information that you provide, such as chassis serial number, MAC address, IP address, and several other possibilities. The **cns** commands are part of the bootstrap configuration of the manufactured device, specified to Cisco Configuration Express at time of order.

Within this scope, Configuration Express and the **cns** commands also allow you to define custom asset tags to your own specifications, which are serialized during manufacture and automatically substituted into the unit’s bootstrap configuration.

Cisco appends tags to the carton for all the various types of IDs supported by the **cns** commands, so that these values can be bar-code read at shipping time and fed back into your systems. Alternatively, these IDs are also available through a direct XML-software interface between your system and the Cisco order-status engine, eliminating the need for bar-code reading. The CNS agent also provides a feedback mechanism whereby the remote device can receive XML events or commands to modify the device’s ID, in turn causing that same device to broadcast an event indicating the old/new IDs.

Management Point

On most networks, a small percentage of individual remote routers get configured locally. This can potentially be a serious problem, not only causing loss of synchronization across your network but also opening your system to the possibility that an automatic reconfiguration might conflict with an existing configuration and cause a router to become unusable or even to lose contact with the network.

To address this problem, you can designate a management point in your network, typically on the Cisco IE2100 CNS configuration engine, and configure it to keep track of the configurations on all remote routers.

To enable this solution, configure the CNS agent to publish an event on the CNS event bus whenever any change occurs to the running configuration. This event indicates exactly what has changed (old/new), eliminating the need for the management point to perform a highly unscalable set of operations such as telnetting into the device, applying a script, reading back the entire running configuration, and determining the difference between old and new configurations. Additionally, you can arrange for Simple Network Management Protocol (SNMP) notification traps of configuration changes occurring through the SNMP MIB set.

Point-to-Point Event Bus

Today’s business environment requires that you be able to ensure your customers a level of service not less than what they are actually paying for. Toward this end, you activate service-assurance applications that broadcast small poll/queries to the entire network while expecting large responses from a typically small subset of devices according to the criteria of the query.

For these queries to be scalable, it is necessary for the replying device to bypass the normal broadcast properties of the event bus and instead reply on a direct point-to-point channel. While all devices need the benefit of the broadcasted poll so that they can all be aware of the query to which they may need to reply, the devices do not have to be aware of each others' replies. Massive copying and retransmission of device query replies, as part of the unnecessary reply broadcast, is a serious scalability restriction.

To address this scalability problem, the CNS event bus has a point-to-point connection feature that communicates directly back to the poller station.

CNS Flow-Through Provisioning provides the following benefits.

Automated Configuration

CNS Flow-Through Provisioning simplifies installation by moving configuration requirements to the CNS configuration engine and allowing the Cisco IOS configuration to update automatically. The registrar uses popular industry standards and technologies such as XML, Active Directory Services Interface (ADSI)/Active Directory, HTTP/Web Server, ATM Switch Processor (ASP), and Publish-Subscribe Event Bus. The CNS configuration agent enables the CNS configuration engine to configure remote routers in a plug-and-play manner.

Unique IP Addresses and Hostname

CNS Flow-Through Provisioning uses DNS reverse lookup to retrieve the hostname by passing the IP address, then assigns the IP address and optionally the hostname to the remote router. Both IP address and hostname are thus guaranteed to be unique.

Reduced Technical Personnel Requirements

CNS Flow-Through Provisioning permits remote routers to be installed by a person with limited or no technical experience. Because configuration occurs automatically on connection to the network, a network engineer or technician is not required for installation.

Rapid Deployment

Because a person with limited or no technical experience can install a remote router immediately without any knowledge or use of Cisco IOS software, the router can be sent directly to its final premises and be brought up without technician deployment.

Direct Shipping

Routers can be shipped directly to the remote end-user site, eliminating warehousing and manual handling. Configuration occurs automatically on connection to the network.

Remote Updates

CNS Flow-Through Provisioning automatically handles configuration updates, service additions, and deletions. The CNS configuration engine performs a push operation to send the information to the remote router.

Security

Event traffic to and from the remote router is opaque to unauthorized listeners or intruders to your network. CNS agents leverage the latest security features in Cisco IOS software.

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:

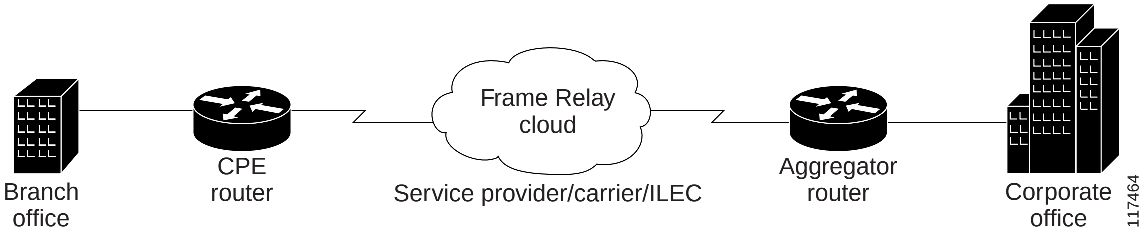
- Ensures consistent CNS commands between Cisco IOS Release 12.3 and 12.3 T.
- Use of a channel service unit (E1 or T1 controller) is allowed.

CNS Frame Relay Zero Touch

The CNS Frame Relay Zero Touch feature provides a CNS zero touch deployment solution over Frame Relay where the CPE router discovers its data-link connection identifier (DLCI) and IP address dynamically, and then contacts a CNS 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 who deploy Frame Relay can create this generic bootstrap configuration without device-specific or network-specific information such as the DLCI, IP address, interface type, controller type (if applicable), or the next hop interface used for the static default route.

Figure 2 illustrates a typical customer network architecture using Frame Relay.

Figure 2 Connectivity in a Frame Relay Customer Network



The CPE router is deployed at multiple sites. Each site connects to a Frame Relay cloud through a point-to-point permanent virtual circuit (PVC). Connectivity from the Frame Relay cloud to the corporate office is through a PVC that terminates at the corporate office. IP traffic sent to the CNS configuration engine is routed through the corporate office. The PVC is identified by its DLCI. The DLCI can vary between branch offices. In order to support zero touch deployment, the CPE router must be able to learn which DLCI to use to connect to the CNS configuration engine.

To support the zero touch capability, the Frame Relay functionality has been modified in the following two ways:

- A new Cisco IOS command, the **ip address dynamic** command has been introduced to discover the CPE router's IP address dynamically based on the aggregator router's IP address. To configure IP over Frame Relay, the local IP address must be configured on the interface.
- The CPE router can now read Local Management Interface (LMI) messages from a Frame Relay switch and determine the list of available DLCIs.

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 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.

When a CPE router is placed in a Frame Relay network, it contains a generic bootstrap configuration. This configuration includes customer-specific Frame Relay configuration (including the LMI type), CNS connect templates, CNS connect profiles, and the **cns config initial** command. This command initiates the CNS connect function.

The CNS connect functionality begins by selecting the first available controller or interface specified by the CNS connect profile and then performs multiple ping iterations through all the associated active DLCIs. 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.

When iterating over the active DLCIs on a Frame Relay interface, the router must be able to automatically go through a list of active DLCIs returned by the LMI messages for that interface and select an active DLCI to use. When more than one of the active DLCIs allow IP connectivity to the CNS configuration engine, the DLCI used will be the first one tried by the CNS connect functionality. If the ping attempt is unsuccessful, the next active DLCI is tried and so on. If connectivity to the CNS configuration engine is unsuccessful for all active DLCIs, the CNS connect function removes the configuration applied to the selected controller or interface, and the CNS connect process restarts with the next available controller or interface specified by the CNS connect profile.

The CNS Frame Relay Zero Touch feature provides the following benefits:

- A service provider can have a single common bootstrap configuration.
- The generic bootstrap configuration does not require the IP address to be hard-wired.
- The point-to-point DLCI does not need to be known in advance.
- IP directly over Frame Relay is allowed.
- Use of a channel service unit (E1 or T1 controller) is allowed.

How to Configure CNS

This section contains the following tasks:

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/univercd/cc/td/doc/product/rtrmgmt/cns/ce/rel13/ag13/index.htm.

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.

Prerequisites

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. **discover** {**line** *line-type* | **controller** *controller-type* | **interface** [*interface-type*]}
   or
   template *name*
9. **exit**
10. **cns config initial** {*host-name* | *ip-address*} [**encrypt**] [*port-number*] [**page** *page*] [**syntax-check**] [**no-persist**] [**source** *ip-address*] [**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.

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	**discover** {**line** *line-type* | **controller** *controller-type* | **interface** [*interface-type*]}
	or
	template *name*
	Example:
	`Router(config-cns-conn)# discover interface serial`
	or
	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** *ip-address*] [**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 the CNS Event and EXEC Agents

Perform this task to enable and configure the CNS Event and EXEC agents.

CNS Event Agent Parameters

The CNS event agent command—**cns event**—has several parameters that can be configured. The **failover-time** keyword is useful if you have a backup CNS event gateway configured. If the CNS event agent is trying to connect to the gateway and it discovers that the route to the backup gateway is available before the route to the primary gateway, the *seconds* argument specifies how long the CNS event agent will continue to search for a route to the primary gateway before attempting to link to the backup gateway.

Unless you are using a bandwidth-constrained link, you should set a keepalive timeout and retry count. Doing so allows the management network to recover gracefully should a Cisco IE2100 configuration engine ever fail. Without the keepalive data, such a failure requires manual intervention on every device. The *seconds* value multiplied by the *retry-count* value determines the length of idle time before the CNS event agent will disconnect and attempt to reconnect to the gateway. We recommend a minimum *retry-count* value of 2.

If the optional **source** keyword is used, the source IP address might be a secondary IP address of a specific interface to allow a management network to run on top of a production network.

Note Although other CNS agents may be configured, no other CNS agents are operational until the **cns event** command is entered because the CNS event agent provides a transport connection to the CNS event bus for all other CNS agents.

SUMMARY STEPS

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

DETAILED STEPS

<table>
<tr><th></th><th>Command or Action</th><th>Purpose</th></tr>
<tr><td>Step 1</td><td><b>enable</b>
<b>Example:</b>
<code>Router enable</code></td><td>Enables privileged EXEC mode.
• Enter your password if prompted.</td></tr>
<tr><td>Step 2</td><td><b>configure terminal</b>
<b>Example:</b>
<code>Router# configure terminal</code></td><td>Enters global configuration mode.</td></tr>
<tr><td>Step 3</td><td><b>cns config partial</b> {<i>host-name</i> | <i>ip-address</i>} [<b>encrypt</b>] [<i>port-number</i>] [<b>source</b> <i>ip-address</i>] [<b>inventory</b>]
<b>Example:</b>
<code>Router(config)# cns config partial 172.28.129.22 80</code></td><td>(Optional) Starts the CNS configuration agent, which provides CNS configuration services to Cisco IOS clients, and initiates an incremental (partial) configuration.
• Use the optional <i>port-number</i> argument to specify the port number for the configuration server. The default is 80.
• Use the optional <b>source</b> keyword and <i>ip-address</i> argument to specify the use of an IP address as the source for CNS configuration agent communications.
• Use the optional <b>inventory</b> keyword to send an inventory of the line cards and modules in the router to the CNS configuration engine as part of the HTTP request.
<b>Note</b> The optional <b>encrypt</b> keyword is available only in images that support SSL.</td></tr>
<tr><td>Step 4</td><td><b>logging cns-events</b> [<i>severity-level</i>]
<b>Example:</b>
<code>Router(config)# logging cns-events 2</code></td><td>(Optional) Enables XML-formatted system event message logging to be sent through the CNS event bus.
• 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).</td></tr>
<tr><td>Step 5</td><td><b>cns exec</b> [<i>host-name</i> | <i>ip-address</i>] [<b>encrypt</b> [enc-<i>port-number</i>]] [<i>port-number</i>] [<b>source</b> <i>ip-address</i>]
<b>Example:</b>
<code>Router(config)# cns exec 10.1.2.3 93 source 172.17.2.2</code></td><td>(Optional) Enables and configures the CNS EXEC agent, which provides CNS EXEC services to Cisco IOS clients.
• Use the optional <i>port-number</i> argument to specify the port number for the EXEC server. The default is 80.
• Use the optional <b>source</b> keyword and <i>ip-address</i> argument to specify the use of an IP address as the source for CNS EXEC agent communications.
<b>Note</b> The optional <b>encrypt keyword</b> is available only in images that support SSL.</td></tr>
</table>

	Command or Action	Purpose
Step 6	**cns event** {*hostname* | *ip-address*} [**encrypt**] [*port-number*] [**backup**] [**failover-time** *seconds*] [**keepalive** *seconds retry-count*] [**source** *ip-address*] [**clock-timeout** *time*] [**reconnect** *time*] **Example:** `Router(config)# cns event 172.28.129.22 source 172.22.2.1`	Configures the CNS event gateway, which provides CNS event services to Cisco IOS clients. • The optional **encrypt** keyword is available only in images that support SSL. • Use the optional *port-number* 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 *seconds* 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 *seconds* and *retry-count* arguments to specify the keepalive timeout in seconds and the retry count. • Use the optional **source** keyword and *ip-address* argument to specify the use of an IP address as the source for CNS event agent communications. • Use the optional **clock-timeout** keyword to specify the maximum time, in minutes, that the CNS event agent will wait for the clock to be set for transports (such as SSL) that require an accurate clock. • Use the optional **reconnect** keyword to specify the configurable upper limit of the maximum retry timeout. **Note** Until the **cns event** command is entered, no transport connections to the CNS event bus are made and therefore no other CNS agents are operational.
Step 7	**exit** **Example:** `Router(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 CNS event agent is connected to the CNS event gateway.
- Use the **show cns event subject** command to check that the image agent subject names are registered. Subject names for the CNS image agent begin with cisco.mgmt.cns.image.

Configuring the CNS Image Agent

Perform this task to configure CNS image agent parameters using CLI commands.

CNS Image Agent ID

CNS uses a unique identifier to identify an image agent associated with that Cisco IOS device. Using the same process as CNS event and configuration agents, the configuration of the **cns id** command determines whether an IP address or MAC address of a specified interface, the hardware serial hardware number of the device, an arbitrary text string, or the hostname of the device is used as the image ID. By default, the system uses the hostname of the device.

The CNS image ID is sent in the content of the messages sent by the image agent and allows an application to know the unique image ID of the Cisco IOS device that generated the message. A password can be configured and associated with the image ID in the image agent messages.

Prerequisites

- To configure the CNS image agent to use HTTP or HTTP over SSL (HTTPS) to communicate with an image server, you need to know the URL for the image server and the URL to which status messages can be sent.
- If you are using HTTPS to communicate with the image server, you must set up security certificates to allow the server to be authenticated by the image agent when the connection is established.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cns id** *type number* {**dns-reverse** | **ipaddress** | **mac-address**} [**event**] [**image**]
   or
   cns id {**hardware-serial** | **hostname** | **string** *text*} [**event**] [**image**]
4. **cns image** [**server** *server-url* [**status s***tatus-url*]]
5. **cns image password** *image-password*
6. **cns image retry** *seconds*
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 id** *type number* {**dns-reverse** | **ipaddress** | **mac-address**} [**event**] [**image**]
	or
	cns id {**hardware-serial** | **hostname** | **string** *text*} [**event**] [**image**]
	Example:
	`Router(config)# cns id fastethernet 0/1 ipaddress image`
	or
	Example:
	`Router(config)# cns id hardware-serial image`
		Specifies a unique CNS ID and interface type and number from which to retrieve the unique ID.
		or
		Specifies a unique CNS ID assigned from the hardware serial number, device hostname, or an arbitrary text string.
		The following information applies to either version of the syntax.
		• Use the **event** keyword to specify an event agent ID.
		• Use the **image** keyword to specify an image agent ID.
		• If no keywords are used, the configuration agent ID is configured.
		Note The **dns-reverse** keyword is not supported in Cisco IOS Release 12.2(33)SRA.

Step 4	**cns image** [**server** *server-url* [**status** *status-url*]] **Example:** `Router(config)# cns image server https://10.21.2.3/cns/imgsvr status https://10.21.2.3/cns/status/`	Enables CNS image agent services and specifies the URL of the image distribution server. • Use the optional **status** keyword and *status-url* argument to specify the URL of a web server to which error messages are written. • If the **status** keyword and *status-url* argument are not specified, status messages are sent as events on the CNS Event Bus. To view the status messages on the CNS Event Bus, the CNS event agent must be configured.
Step 5	**cns image password** *image-password* **Example:** `Router(config)# cns image password abctext`	(Optional) Specifies a password for CNS image agent services. • If a password is configured, the password is included with the image ID in CNS image agent messages sent out by the image agent. The receiver of these messages can use this information to authenticate the sending device.

	Command or Action	Purpose

Step 6	`cns image retry` *`seconds`*
	Example:
	`Router(config)# cns image retry 240`
		(Optional) Specifies an image upgrade retry interval in seconds.
		• The default interval is 60 seconds.

Step 7	`exit`
	Example:
	`Router(config)# exit`
		Exits global configuration mode and returns the router to privileged EXEC mode.

What to Do Next

Proceed to the "Retrieving a CNS Image from a Server" section to connect to the web server and download an image.

If any of the commands in the task fail, proceed to the "Troubleshooting CNS Agents" section to try to determine the problem.

Configuring CNS Security Features

Perform this task to configure CNS security features.

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.

CNS Security Enhancement

CNS messages can be configured to use the CNS SOAP message structure, in which the username and password are authenticated. If AAA is configured, then CNS SOAP messages will be authenticated with AAA. If AAA is not configured, there will be no authentication.

Use the **cns aaa authentication** command to determine whether the CNS messages are using AAA security or not. If the **cns aaa authentication** command is configured, then all incoming SOAP messages into the device are authenticated by AAA.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cns trusted-server** {**all-agents** | **config** | **event** | **exec** | **image**} {*host-name* | *ip-address*}

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**} {*host-name* | *ip-address*}
	Example:
	`Router(config)# cns trusted-server event 10.19.2.5`
		Configures a CNS trusted server for the specified hostname or IP address.

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.

Retrieving a CNS Image from a Server

Perform this task to poll the image distribution server using HTTP or HTTPS.

Prerequisites

This task assumes that you have already configured the CNS image agent using the tasks in the "Configuring the CNS Image Agent" section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cns image retrieve** [**server** *server-url* [**status** *status-url*]]

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 image retrieve** [**server** *server-url* [**status** *status-url*]] **Example:** Router(config)# cns image retrieve server https://10.19.2.3/imgsvr/ status https://10.19.2.3/imgsvr/status/	Contacts a Cisco CNS image distribution server and downloads a new image if a new image exists. • Use the optional **status** keyword and *status-url* argument to specify the URL of a web server to which status messages are written. • If the **server** and **status** keywords are not specified, the server and status URLs configured with the **cns image** command are used. **Note** We recommend using the **cns trusted-server** command to specify the host part of the server or status URL as a trusted server.

Troubleshooting Tips

- If the web server appears to be down, use the **ping** command to check connectivity.
- If using HTTP, use the **show ip http client all** command to display information about HTTP clients and connections.

Retrieving a CNS 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).

Prerequisites

This task assumes that you have specified a trusted server using tasks in the "CNS Security Enhancement" section.

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** *ip-address*] [**status** *url*] [**event**] [**inventory**]

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 config retrieve** {*host-name* | *ip-address*} [**encrypt**] [*port-number*] [**page** *page*] [**overwrite-startup**] [**retry** *retries* **interval** *seconds*] [**syntax-check**] [**no-persist**] [**source** *ip-address*] [**status** *url*] [**event**] [**inventory**] **Example:** `Router(config)# cns config retrieve server1 retry 5 interval 45`	Allows the router to retrieve configuration data from a web server. - 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.

Troubleshooting Tips

If you need to stop the retrieval process, enter the Ctrl+Shift+6 key sequence.

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

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.

Prerequisites

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. Else (use above text?)

Restrictions

- 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:** `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	**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* [**user** *username*] {**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.

	Command or Action	Purpose
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.
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 notification 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.

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.

Troubleshooting CNS Agents

This section explains how to troubleshoot CNS agent issues.

The **show** commands created for the CNS 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 CNS 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 CNS 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:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. - Enter your password if prompted.
Step 2	**show cns image status** **Example:** `Router# show cns image status`	(Optional) Displays information about the CNS image agent status.
Step 3	**clear cns image status** **Example:** `Router# clear cns image status`	(Optional) Clears CNS image agent status statistics.
Step 4	**show cns image connections** **Example:** `Router# show cns image connections`	(Optional) Displays information about CNS image management server HTTP or HTTPS connections.
Step 5	**show cns image inventory** **Example:** `Router# show cns image inventory`	(Optional) Displays inventory information about the CNS 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:** `Router# debug cns image all`	(Optional) Displays debugging messages for CNS image agent services.
Step 7	**show cns event connections** **Example:** `Router# show cns event connections`	(Optional) Displays the status of the CNS 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:** `Router# show cns event subject subject1`	(Optional) Displays a list of subjects of the CNS event agent that are subscribed to by applications.

Examples

This section provides the following output examples:

- Sample Output for the show cns image status Command
- Sample Output for the show cns image connections Command
- Sample Output for the show cns image inventory Command

- Sample Output for the debug cns image Command

Sample Output for the show cns image status Command

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

```
Router# 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
```

Sample Output for the show cns image connections Command

In the following example, information about the status of the CNS 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
```

Sample Output for the show cns image inventory Command

In the following example, information about the CNS 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
.
```

Sample Output for the debug cns image Command

In the following example, debugging messages for all CNS image agent services are displayed using the **debug cns image** privileged EXEC command. The CNS 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 IOS device.

```
debug cns image all

All cns image debug flags are on

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
```

Sample Output for the show cns event Commands

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

```
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 CNS event agent that are subscribed to by applications:

```
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 CNS

This section provides the following configuration examples:

- Deploying the CNS Router: Example, page 39
- Configuring a Partial Configuration: Example, page 39
- Enabling and Configuring CNS Agents: Example, page 39
- CNS Flow-Through Provisioning: Examples, page 41
- Command Scheduler Policy Lists and Occurrences: Examples, page 44
- Retrieving a CNS Image from a Server: Example, page 44
- Retrieving a CNS Configuration from a Server: Examples, page 45
- Using the CNS Zero Touch Solution: Examples, page 45

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
```

Configuring a Partial Configuration: Example

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.

The following example shows incremental (partial) configuration on a remote router. The CNS configuration engine IP address is 172.28.129.22, and the port number is 80.

```
 cns config partial 172.28.129.22 80
```

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 exec agent is enabled with an IP address of 172.28.129.23, and 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 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 CNS image agent parameters are configured using the CLI. An image ID is specified to use the IP address of the FastEthernet interface 0/1, a password is configured for the CNS image agent services, the CNS image upgrade retry interval is set to four minutes, and image management and status servers are configured.

```
cns id FastEthernet0/1 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 CNS image agent is configured to use the CNS Event Bus. An image ID is specified as the hardware serial number of the networking device, the CNS event agent is enabled with a number of parameters, and the CNS image agent is enabled without any keywords or options. The CNS image agent will listen for events on the CNS 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
```

CNS Flow-Through Provisioning: Examples

Cisco Configuration Express File Using T1 over HDLC Protocol Example

The following example shows use of the Cisco Configuration Express file to configure the remote router before delivery to its final premises. In the example, 172.28.129.22 is the IP address of the CNS configuration engine.

```
cns config initial 172.28.129.22 no-persist
!cns configure and event agents
cns event 172.28.129.22
controller t1 0
!T1 configuration
framing esf
linecode b8zs
channel-group 0 timeslots 1-24 speed 64
exit
cns id s0:0 ipaddress
interface s0:0
!Assigns IP address to s0:0
ip address slarp retry 2
exit
ip route 10.0.0.0 0.0.0.0 s0:0
!IP static route
end
```

T1 Configuration Template Example

The following example shows use of the T1 configuration template to build the configuration for use on T1:

```
hostname ${LDAP://this:attrName=IOShostname}
enable password ${LDAP://this:attrName=IOSpassword}
controller T1 0
clock source ${LDAP://this:attrName=IOST1-clocksource}
linecode ${LDAP://this:attrName=IOST1-line}
framing ${LDAP://this:attrName=IOST1-framing}
channel-group ${LDAP://this:attrName=IOST1-channel-group}
timeslots ${LDAP://this:attrName=IOST1-timeslots}
speed ${LDAP://this:attrName=IOST1-speed}
```

Voice Configuration Template Example

The following example shows use of the voice configuration template to build the configuration for using voice:

```
voice-port 1/1
codec ${LDAP://this:attrName=IOSvoice-port1}
exit
dial-peer voice 1 pots
application ${LDAP://this:attrName=IOSdial-peer1}
port 1/1
```

Remote Router Example

The following example shows a remote router configuration:

```
 show running-config

Current configuration: 1659 bytes
!
version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname tira-24V
!
!
network-clock base-rate 64k
ip subnet-zero
ip cef
!
ip audit notify log
ip audit po max-events 100
!
class-map match-any voice
match access-group 100
!
!
policy-map qos
class voice
priority percent 70
voice service voip
h323
!
no voice confirmation-tone
voice-card 0
!
!
controller T1 0
framing sf
linecode ami
!
controller T1 1
mode cas
framing esf
linecode b8zs
ds0-group 0 timeslots 1 type e&m-immediate-start
ds0-group 1 timeslots 2 type e&m-immediate-start
!
!
interface Ethernet0
ip address 10.1.1.2 255.255.0.0
!
interface Serial0
bandwidth 1536
ip address 10.11.11.1 255.255.255.0
no ip mroute-cache
load-interval 30
clockrate 148000
!
ip classless
ip route 223.255.254.254 255.255.255.0 10.3.0.1
!
no ip http server
ip pim bidir-enable
!
access-list 100 permit udp any range 16384 32767 any
access-list 100 permit tcp any any eq 1720
call rsvp-sync
!
voice-port 1:0
timeouts wait-release 3
!
voice-port 1:1
timeouts wait-release 3
```

```
!
!
mgcp profile default
!
dial-peer cor custom
!
dial-peer voice 1000 pots
destination-pattern 1000
port 1:0
forward-digits 0
!
dial-peer voice 1001 pots
destination-pattern 1001
no digit-strip
port 1:1
forward-digits 0
!
dial-peer voice 2000 voip
destination-pattern 2000
session target ipv4:10.11.11.2
codec g711ulaw
!
dial-peer voice 2001 voip
destination-pattern 2001
session target ipv4:10.11.11.2
signal-type ext-signal
codec g711ulaw
!
!
line con 0
line aux 0
line 2 3
line vty 0 4
```

The following example shows configuration of a serial interface to connect to and download a configuration from a Cisco IE2100 CNS configuration engine. The IE2100 IP address is 10.1.1.1. The gateway IP address to reach the 10.1.1.0 network is 10.11.11.1. The CNS default ID is the hostname, so that **cns id** command is not needed. However, the **hostname** command is key to retrieving the configuration file on the CNS configuration engine.

This configuration auto-tries ever serial interface on the remote router in turn, applies the **config-cli** commands to that interface, and tries to ping the address in the **cns config initial** command. When it succeeds, it performs a normal initial configuration.

```
! Initial basic configuration (serial interface) PPP
cns connect serial retry-interval 1 retries 1
config-cli ip address negotiated
config-cli encapsulation ppp
config-cli ip directed-broadcast
config-cli no keepalive
config-cli no shutdown
exit
hostname 26ML
ip route 10.1.1.1 255.255.255.0 10.11.11.1
cns config initial 10.1.1.1 no-persist
cns inventory config
! Initial basic configuration (serial interface) HDLC
cns config connect serial retry-interval 1 retries 1
config-cli ip address slarp retry 1
config-cli no shutdown
exit
hostname tira-36V
ip route 10.1.1.1 255.255.255.0 10.11.11.1
```

```
cns config initial 10.1.1.1 no-persist
cns inventory config
Incremental configuration (serial interface)
cns config partial 10.1.1.1
cns event 10.1.1.1
```

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 Image from a Server: Example

In the following example, the CNS image agent polls a file server using the **cns image retrieve** command. Assuming that the CNS 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/
```

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 CNS.

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*, Release 12.4T
CNS Configuration Engine	*Cisco Intelligence Engine 2100 Configuration Registrar Manual*, Release 1.1 or later *Cisco CNS Configuration Engine Administrator's Guide*
IAD and Router Hardware and Software	Cisco IAD2420 series hardware and software documents, available online at http://www.cisco.com/univercd/cc/td/doc/product/access/iad/iad2420/index.htm Cisco 2600 series hardware and software documents, available online at http://www.cisco.com/univercd/cc/td/doc/product/access/acs_mod/cis2600/hw_inst/index.htm Cisco 3600 series hardware and software documents, available online at http://www.cisco.com/univercd/cc/td/doc/product/access/acs_mod/cis3600/hw_inst/index.htm

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 release.	To locate and download MIBs for selected platforms, Cisco IOS 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

Table 2 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(1), 12.0(3)S or a later release appear in the table.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 2 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 2 Feature Information for CNS

Feature Name	Releases	Feature Information
CNS	12.2(25)S 12.2(33) SRA	The 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. The following sections provide information about this feature: • Prerequisites for CNS, page 2 • Restrictions for CNS, page 2 • Information About CNS, page 3 • CNS, page 4 • CNS Configuration Agent, page 4 • How to Configure CNS, page 17 • Configuration Examples for CNS, page 39 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**.

Table 2 Feature Information for CNS (continued)

Feature Name	Releases	Feature Information
CNS Configuration Agent	12.0(18)ST 12.0(22)S 12.2(2)T 12.2(33)SRA	The CNS Configuration Agent feature supports routing devices by providing the following: • Initial configurations • Incremental (partial) configurations • Synchronized configuration updates The following sections provide information about this feature: • CNS Configuration Agent, page 4 • Initial CNS Configuration, page 4 • Incremental CNS Configuration, page 5 • Synchronized Configuration, page 5 • Configuring the CNS Event and EXEC Agents, page 20 • Troubleshooting CNS Agents, page 34 The following commands were introduced or modified by this feature: **cns config cancel**, **cns config initial**, **cns config partial**, **cns config retrieve**, **debug cns config**, **debug cns xml-parser**, **show cns config outstanding**, **show cns config stats**, **show cns config status**.
CNS Config Retrieve Enhancement with Retry and Interval	12.4(15)T	The Cisco Networking Services (CNS) Config Retrieve Enhancement with Retry and Interval feature add 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 sections provide information about this feature: • 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**

Table 2 Feature Information for CNS (continued)

Feature Name	Releases	Feature Information
CNS Enhanced Results Message	12.2(33)SRA 12.4(4)T	The CNS Enhanced Results Message feature sends a second CNS result message to the subject "cisco.cns.config.results" in addition to the CNS results messages sent to the CNS Event bus after a partial configuration is complete. The following sections provide information about this feature: • CNS Results Messages, page 6 • Configuring the CNS Event and EXEC Agents, page 20 • Configuring a Partial Configuration: Example, page 39 The following command was modified by this feature: **cns config partial**.
CNS Event Agent	12.0(18)ST 12.0(22)S 12.2(2)T 12.2(33)SRA	The CNS Event Agent is part of the Cisco IOS infrastructure that allows Cisco IOS applications to publish and subscribe to events on a CNS Event Bus. CNS Event Agent works in conjunction with the CNS Configuration Agent feature. The following sections provide information about this feature: • CNS Event Agent, page 5 • Configuring the CNS Event and EXEC Agents, page 20 • Troubleshooting CNS Agents, page 34 The following commands were introduced or modified by this feature: **cns event**, **show cns event connections**, **show cns event stats**, **show cnsevent subject**.

Table 2 Feature Information for CNS (continued)

Feature Name	Releases	Feature Information
CNS Flow-Through Provisioning	12.2(2)T 12.2(2)XB 12.2(11)YT 12.2(11)YV	Cisco Networking Services (CNS) Flow-Through Provisioning provides the infrastructure for automated configuration of large numbers of network devices. Based on CNS event and config agents, it eliminates the need for an onsite technician to initialize the device. The result is an automated workflow from initial subscriber-order entry through Cisco manufacturing and shipping to final device provisioning and subscriber billing. This focuses on a root problem of today's service-provider and other similar business models: use of human labor in activating service. The following sections provide information about this feature: • Prerequisites for CNS, page 2 • CNS Flow-Through Provisioning, page 11 • Configuring the CNS Event and EXEC Agents, page 20 • CNS Flow-Through Provisioning: Examples, page 41 The following commands were introduced or modified by this feature: **cns config cancel**, **cns config connect-intf**, **cns config initial**, **cns config partial, cns config notify**, **cns event**, **cns id**, **cns inventory**, **cns mib-access encapsulation**, **cns notifications encapsulation**, **config-cli**, **debug cns config**, **debug cns event**, **debug cns management**, **debug cns xml-parser**, **line-cli, show cns config connections**, **show cns config outstanding**, **show cns event stats**, **show cns event subject**. **Note** The **cns config connect-intf** command was replaced by the **cns connect** and **cns template connect** commands in Cisco IOS Release 12.3(2)XF, 12.3(8)T, 12.3(9), 12.2(33)SRA, 12.2(31)SB2, and later releases. **Note** The **config-cli** and **line-cli** commands were replaced by the **cli (cns)** command in Cisco IOS Release 12.3(2)XF, 12.3(8)T, 12.3(9), 12.2(33)SRA, 12.2(31)SB2, and later releases.

Table 2 Feature Information for CNS (continued)

Feature Name	Releases	Feature Information
CNS Frame-Relay Zero Touch	12.3(2)XF 12.3(8)T	The CNS Frame Relay Zero Touch feature provides a CNS zero touch deployment solution over Frame Relay where the CPE router discovers its DLCI and IP address dynamically and then contacts a CNS engine to retrieve its full configuration automatically. The following sections provide information about this feature: • Restrictions for CNS, page 2 • CNS Frame Relay Zero Touch, page 15 • Deploying the CNS Router, page 17 • Using the CNS Zero Touch Solution: Examples, page 45 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 in Cisco IOS Releases 12.3(2)XF, 12.3(8)T, 12.3(9), 12.2(33)SRA, 12.2(31)SB2, and later releases. **Note** The **config-cli** and **line-cli** commands were replaced by the **cli (cns)** command in Cisco IOS Releases 12.3(2)XF, 12.3(8)T, 12.3(9), 12.2(33)SRA, 12.2(31)SB2, and later releases.

Table 2 Feature Information for CNS (continued)

Feature Name	Releases	Feature Information
CNS Image Agent	12.2(33)SEE 12.3(1) 12.2(31)SB2 12.2(33)SRB	The CNS Image Agent feature is an infrastructure in Cisco IOS software to enable automated installation and activation of Cisco IOS images on Cisco IOS networking devices. The following sections provide information about this feature: • Prerequisites for CNS, page 2 • Restrictions for CNS, page 2 • CNS Image Agent, page 5 • Configuring the CNS Image Agent, page 24 • Retrieving a CNS Image from a Server, page 27 • Troubleshooting CNS Agents, page 34 • Enabling and Configuring CNS Agents: Example, page 39 • Retrieving a CNS Image from a Server: Example, page 44 The following commands were introduced or modified by this feature: **clear cns image connections, clear cns image status, cns id, cns image, cns image password, cns image retrieve, cns image retry, debug cns image, show cns image connections, show cns image inventory, show cns image status**.
CNS Interactive CLI	12.0(28)S 12.2(18)SXE 12.2(18)SXF2	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. The following section provides information about this feature: • CNS Interactive CLI, page 10
CNS Security Enhancement	12.4(9)T 12.2(33)SRA	The CNS Security Enhancement feature improves the security of Cisco Networking Services (CNS) messages by authenticating sender credentials through the use of the Service-Oriented Access Protocol (SOAP) message format. The following sections provide information about this feature: • CNS Message Formats, page 6 • CNS Security Enhancement, page 9 • Configuring CNS Security Features, page 26 The following commands were introduced or modified by this feature: **cns aaa authentication, cns message format notification**.

Table 2 Feature Information for CNS (continued)

Feature Name	Releases	Feature Information
CNS Zero Touch	12.3(9)	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 sections provide information about this feature: • Prerequisites for CNS, page 2 • Restrictions for CNS, page 2 • CNS Zero Touch, page 15 • Deploying the CNS Router, page 17 • Using the CNS Zero Touch Solution: Examples, page 45 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 in Cisco IOS Release 12.3(2)XF, 12.3(8)T, 12.3(9), 12.2(33)SRA, 12.2(31)SB2, and later releases. **Note** The **config-cli** and **line-cli** commands were replaced by the **cli (cns)** command in Cisco IOS Release 12.3(2)XF, 12.3(8)T, 12.3(9), 12.2(33)SRA, 12.2(31)SB2, and later releases.
Command Scheduler	12.3(1) 12.2(33)SRA	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 sections provide information about this feature: • Restrictions for CNS, page 2 • Command Scheduler, page 10 • Configuring Command Scheduler Policy Lists and Occurrences, page 29 • Command Scheduler Policy Lists and Occurrences: Examples, page 44 The following commands were introduced or modified by this feature: **cli**, **debug kron**, **kron occurrence**, **kron policy-list**, **policy-list**, **show kron schedule**.

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