Deploying Network Changes

This chapter references `$WAE_HOME`, which is the directory in which the packages are installed. The default `$WAE_HOME` is `/opt/cariden/software`.

When WAE Automation software is installed, the following packages are installed in `$WAE_HOME`:

- wae-core—Contains WAE Core server files. It also contains configuration files that enable the use of WAE Core REST and Thrift APIs.
- wae-db—Contains WAE Core database files.
- wae-messaging—WAE messaging system that uses Java Message Service (JMS).
- wae-osc—Contains configuration files for Cisco Open SDN Controller (OSC).
- wae-appenginecore and wae-designapiserver—Services that enable the use of the WAE Design REST API, the Dynamic SLA Management API, and the Stage Management REST API.

WAE Core Server

Note The configuration instructions in this chapter are for single-system environments only. For high-availability deployments, contact your support representative.

WAE Core Configuration Files

The `$WAE_HOME/wae-core/etc` directory contains the following configuration files with options that may be configured (see Table 4-1).

Most of the configurations mentioned here are set to default values but are commented out. For example, if you want to enable authentication, simply uncomment the entry #authenticationEnabled=true in the appropriate file.

Note Only the most common configuration files are listed in Table 4-1.

Table 4-1 WAE Core Configuration Files

Configuration File	Description
com.cisco.wano.nsps.demand.persistimpl.cql.cfg	Contains demand persistence configurations. For example, interval size of evaluated projected plans for bandwidth calendering.
com.cisco.wano.nsps.deployer.cfg	Contains non-PCEP deployer configurations. For example, type of deployment to use and timeout period for shutting down routes.
com.cisco.wano.nsps.deployer.ncs.cfg	Contains Network Services Orchestrator (NSO) deployer configurations.

com.cisco.wano.nsps.nbrs.cfg	Contains northbound RESTful API configurations.
	To manage the behavior of the REST northbound API, set these properties:
	Note Several of them increase security for accessing the APIs by enabling authentication, changing the credentials, and SSL port.
	• To enable authentication, change the authenticationEnabled property to true.
	`authenticationEnabled=true`
	• To change the username and password credentials, use these properties.
	`username=<username>`
	`password=<password>`
	• To configure the protocol, REST service port, and SSL port, follow these guidelines. If neither HTTP, nor HTTPS is set, HTTPS is the default.
	• If receiving timeout errors, increase the timeout value.
	`nbQSendOptions=?requestTimeout=<# of milliseconds>`

com.cisco.wano.nsps.thrift.cfg	Contains northbound THRIFT API configurations. It contains the following configurable options:
	• Enable or disable the Thrift northbound API by setting the thriftEnabled property.
	`thriftEnabled=<true/false>`
	• Set the port on which Thrift listens. The default port is 9898.
	`port=<port_number>`
	• If receiving timeout errors, increase the timeout value.
	`nbQSendOptions=?requestTimeout=<# of milliseconds>`

Configuration File	Description
com.cisco.wano.nsps.deployer.pcep.cfg	Contains PCEP deployer configurations.
com.cisco.wano.nsps.engine.cfg	Contains NSPS engine configurations. You can configure the following: • location of plan files • changing default port, • processing threads • number of projected plans for bandwidth calendaring • projection reload configurations. The WAE API starts a new process when it invokes a WAE tool. The number of concurrent WAE tool invocations is controlled by the number of WAE threads. Tuning these parameters is dependent not only on the number of processors, but also on other applications that might be running on the device. As a best practice, set to 4 for devices that have 16 GB of memory and to 8 for devices that have 32 GB of memory. Setting the procThreads property determines how much multiprocessing occurs and can improve performance. The default is set to 8. com.cisco.nsps.engine.procThreads=<#> The following is a list of other parameters available: • To control number of projected plans for bandwidth calendaring: com.cariden.nac.service.projection.projectionSize=20 • To allow or deny admission of demands that do not fit into bandwidth calendaring projection window,: com.cariden.nac.service.projection.allowDemandSkew=false • To configure location to which plan files can be dropped (default is $WAE_HOME/plans): com.cisco.wano.nsps.engine.plan.dropFilesBase= The dropped plan file will be autoloaded. • To configure location to which uploaded plan files will be stored temporarily until finished processing (default is ${java.io.tmpdir} or /tmp if the former is not defined): com.cisco.wano.nsps.engine.plan.uploadFilesBase= • To configure port on which local SSH server is accepting connections (default is 22): com.cisco.wano.nsps.engine.sshServicePort=22 This is used to upload plan files over SCP.

Memory

Configuration file: `$WAE_HOME/wae-core/bin/setenv`

If you encounter a memory error, increase the WAE process memory. In this example, these are set to a minimum of 4G and a maximum of 10G.

```
if [ -z $JAVA_MIN_MEM ]; then
 export JAVA_MIN_MEM=4G
fi
if [ -z $JAVA_MAX_MEM ]; then
 export JAVA_MAX_PERM_MEM=10G
fi
```

Logging

Configuration file: `$WAE_HOME/wae-core/etc/org.ops4j.pax.logging.cfg`

By default, log file size limit is 10 MB. Each time a log file reaches that limit, it is copied to a file named nspsmix.log.#. Each time a new log file is created, the number of each existing log file is increased by one. The newest log file, however, does not receive a number. For example, if you had nspxmix.log.1 through nspxmix.log.5, the one without a number would be the most recent, the one ending in 1 would be the second most recent, and the one ending in 5 would be the oldest. By default, the maximum number of backup log files is 10.

Property	Default	Description
log4j.logger.com.cisco=<log_level> **Example:** log4j.logger.com.cisco=TRACE	DEBUG	The type of log level to use can be ERROR, WARN, INFO, DEBUG, or TRACE.
log4j.appender.out.maxFileSize=<whole_number>[MB | GB]	10MB	Maximum permissible log file size.
log4j.appender.out.maxBackupIndex=<whole_number>	10	Maximum permissible number of backup log files.

Deployer Module

The WAE Deployer pushes RSVP or SR LSP create, modify or delete requests to either the Cisco Open SDN Controller (OSC) or Cisco Network Services Orchestrator (NSO). OSC and NSO then perform the requested operations on the network.

- If an LSP is PCEP, OSC is used to manage the PCEP initiated or PCEP delegated LSP.
- If an LSP is not PCEP, NSO is used to change the device configuration.

For successful deployment, the following criteria must be met:

- An LSP in the WAE network model must have an LSP path.
- The LSP must be explicitly routed for RSVP or the LSP path segment list must be defined for segment routing.

The default settings are configured so that each LSP type (PCEP and non-PCEP) is correctly deployed using either OSC or NSO:

- $WAE_HOME/wae-core/etc/com.cisco.wano.nsps.deployer.cfg—For non-PCEP deployments.
- $WAE_HOME/wae-core/etc/com.cisco.wano.nsps.deployer.pcep.cfg—For PCEP deployments.

Note The configuration instructions in this chapter are for single-system environments only. For high-availability deployments, contact your support representative.

Deploying LSPs Using OSC

In $WAE_HOME/wae-core/etc/com.cisco.wano.nsps.deployer.pcep.cfg, verify that OSC will be used for PCEP LSPs sent to WAE. This is the default setting:

```
pcepDeployerProxy=odlPcepDeployerProxy
```

Note For more options that you can set, see Table 4-2.

Table 4-2 com.cisco.wano.nsps.deployer.pcep.cfg Options

Option	Description
Handling of deployment failures	
deployerFailurePolicy=BEST_EFFORT	Once the failure occurs, continues to deploy as much as possible. To determine the deployment state, use the following API. /wae/network/deployer/job/jobState
deployerFailurePolicy=STOP_ON_FAILUREST OP_ON_FAILURE	Stops the deployment immediately upon failure, and nothing is deployed.
Configuring proxy	
pcepDeployerProxy=testPcepDeployerProxy	Invokes the PCEP Deployer, but does not communicate with the OSC controller. This is the default value.
pcepDeployerProxy=odlPcepDeployerProxy	Invokes the PCEP Deployer using this proxy. You must set this parameter with this option if using OSC to discover PCEP tunnels.

Deploying LSPs Using Cisco NSO

Before You Begin

- Obtain the Network Element Drivers (NED) for each device vendor.
- Obtain the traffic engineering service.
- NSO must be installed. The default settings assume WAE and NSO are installed on the same machine using the NSO default login and port. If NSO is installed on a different machine or the login or port have been changed, update the WAE configuration file /opt/cariden/software/wae-core/etc/com.cisco.wano.nsps.deployer.ncs.cfg with the appropriate information.

Note NSO installation is outside the scope of this document. Please contact a support representative if you need the NEDs and the traffic engineering service.

Step 1 In $WAE_HOME/wae-core/etc/com.cisco.wano.nsps.deployer.cfg, verify that NSO will be used for non-PCEP LSPs sent to WAE. This is the default setting:

```
nonPcepDeployer=ncs
```

Step 2 (Optional and only with NSO 3.4) To populate the NSO device list from the plan file and the auth.enc authentication file, issue the add_nodes_to_nso command. This WAE CLI tool only works with NSO version 3.4.

```
add_nodes_to_nso -plan-file <filename> -nso-server <address>
```

where

- <filename>—Input plan file name (.pln/.txt)
- <address>—NSO server address

For example,

```
add_nodes_to_nso -plan-file /opt/cariden/work/pce-test.pln -nso-server localhost
```

Note If devices use Telnet, edit the auth.enc file (encrypted) so that it uses Telnet instead of SSH (default).

a. Add a new column named `Protocol` with the value `telnet`.

Note Ensure that the auth.enc file remains tab-delimited. There cannot be any spaces, only tabs, between each entry in the file.

For example,

```
<UserTable>
IPRegexp    Username    Password   EnablePassword   Protocol
            cisco       cisco      cisco            telnet
```

Enabling BPL-LS Collection Within OSC

Step 1 To enable OSC to use BGP-LS, you must configure a BGP-LS session between one router in the IGP and OSC. Edit the following lines with the appropriate values for your network and server:

Note Sometimes, you need to start, then stop OSC to initially create the configuration files if they don't exist.

a. $WAE_HOME/wae-osc/etc/opendaylight/karaf/41-bgp-example.xml

1. Uncomment the section beginning at line 68.
2. Change the appropriate values for host, local-as, bgp-id, and iana-linstate-attribute-type:

   `<host>10.10.14.27</host>`—Enter the IP address of the BGP-LS speaking router

`<local-as>65000</local>`—Set AS as the same AS on the router that OSC is iBPG peers with

`<bgp-id>192.172.143.8</bgp-id>`—Enter the local OSC server interface IP address that will be used as the source for the BGP session

b. $WAE_HOME/wae-osc/etc/opendaylight/karaf/31-bgp-example.xml

1. Change the iana-listat-attribute value to true:

```
<iana-linkstate-attribute-type>true</iana-linkstate-attribute-type>
```

Step 2 Restart the OSC service.

```
service wae-osc restart
```

Verifying LSP Deployment

Step 1 Configure LSP network changes (for example, create a tunnel) using APIs or WAE Design.

Note The LSPs must have explicitly routed paths.

Step 2 Deploy the plan file.

Step 3 Execute the following APIs and check if the job status was successful or failed:

- /network/deployer/job/details
- /network/deployer/job/jobState