Network Interface Modules (NIMOs)

The following sections describe how to configure different types of network collection and other NIMO capabilities. Although the following topics show how to use the Expert Mode for configuration, you can also use the WAE UI or WAE CLI. The topics describe the options that can be configured using any interface.

NIMO Descriptions

Each NIMO has capabilities (derived from NETCONF protocol capabilities) that determine what it collects or deploys. The following table lists a description of each NIMO.

To list the capabilities of each NIMO, click the **get-capabilities** button (in the Expert Mode) after a NIMO is configured.

Note If you wish to consolidate different data collections (NIMO collections) under a single network model, configure the aggregator before running any collections. For more information, see NIMO Collection Consolidation, on page 9.

Collection or Capability	NIMO	Description	Prerequisite/Notes
Network Collection NIMOs			
IGP Topology Collection, on page 4	topo-igp-nimo	Discovers IGP topology using login and SNMP.	This is a basic topology collection (topology NIMO). The resulting network model is used as the source network for other NIMOs.
BGP-LS Topology Collection Using XTC, on page 6	topo-bgpls-xtc-nimo	Discovers Layer 3 topology using BGP-LS via XTC. It uses raw XTC data as the source for the topology. Node and interface/port properties are discovered using SNMP.	• XTC agents must be configured before running this collection. See Configuring XTC Agents Using the Expert Mode. • This is a basic topology collection for networks using XTC. The resulting network model is used as the source network for other NIMOs.
VPN Collection, on page 12	topo-vpn-nimo	Discovers Layer 2 and Layer 3 VPN topology.	A network model with basic topology collection must exist.
BGP Peer Collection, on page 17	topo-bgp-nimo	Discovers BGP peering using login and SNMP.	A network model with basic topology collection must exist.
LAG Ports and LMP Interface Collection, on page 16	port-cfg-parse-nimo	Discovers LAG ports and Link Management Protocol (LMP) interfaces from router configurations in the network.	• A network model with basic topology collection must exist. • A Configuration Parsing agent must be configured before running this collection. See Configuring the Configuration Parsing Agent.
Configure Multi-Layer Collection	optical-nimo	In conjunction with other NIMOs, the final network collection discovers Layer 1 (optical) and Layer 3 topology.	There are configurations that must take place before configuring the optical-nimo. See Multi-Layer Collection Workflow .
LSP Collection Using NSO NEDs, on page 13	lsp-config-nimo	Discovers LSPs using NEDs and the LSP binding SIDs via NETCONF.	A network model with basic topology collection must exist.
LSP Collection Using SNMP, on page 19	lsp-snmp-nimo	Discovers LSPs using SNMP.	A network model with basic topology collection must exist.
PCEP LSP Collection Using XTC, on page 16	lsp-pcep-xtc-nimo	Discovers PCEP LSPs using XTC.	The BGP-LS Topology Collection Using XTC, on page 6 must be completed before running this collection.
Segment Routing LSP Traffic Collection, on page 20	sr-traffic-matrix-nimo	Discovers SR LSP traffic information.	• A network model with basic topology collection must exist. • Telemetry must be set up on the router.

Collection or Capability	NIMO	Description	Prerequisite/Notes
NIMO Collection Consolidation, on page 9	—	Aggregates various NIMO information into a single consolidated network model.	Configured network models with information you want to merge into one final network model.
Autonomous System (AS) Model Consolidation, on page 11	as-merger	Resolves interfaces, circuits, etc., that are shared between AS models to create a single, consolidated network model.	• Confirm that collection has been completed on the individual AS network models that you want to merge. • Any AS network models that use the topo-bgpls-xtc NIMO must each have an Autonomous System Number (ASN) assigned to it.
Additional NIMOs			
Continuous Collection, on page 20	traffic-poll-nimo	Collects traffic statistics (interface measurements) using SNMP polling.	• A network model with basic topology collection. • If collecting LSP traffic, a network model with LSP collection must exist. See LSP Collection Using SNMP, on page 19. • If collecting VPN traffic, a network model with VPN collection must exist. See VPN Collection, on page 12.
Network Model Visualization, on page 23	layout-nimo	Adds layout properties to a source model to improve visualization.	• A consolidated network model (aggregator). • After the layout-nimo is configured, a plan file containing layout properties must be imported back into the layout-nimo model.
Demand Mesh Creation, on page 25	demandmesh-creator-nimo	Creates a demand mesh between a set of source and destination nodes.	A network model with basic topology collection must exist.
Demand Deduction, on page 24	demand-deduction-nimo	Runs demand deduction using measured SNMP traffic and by applying demand mesh builds of end-to-end demands (traffic matrix) to the network model.	This NIMO requires that you use a consolidated network model (aggregator) as a source network that includes demands and interface measurements (traffic-poller).
Running External Scripts Against a Network Model, on page 26	external-executable-nimo	Runs customized scripts to append additional data to a source network model.	A source network model and a custom script.

Basic Topology Collection

The network model resulting from basic topology collections (topology NIMOs) is used as the source network for additional data collections. To consolidate topology and other data collections, you must first set up the aggregator before running any collection. For more information on the aggregator, see NIMO Collection Consolidation, on page 9.

IGP Topology Collection

The IGP topology (topo-igp-nimo) discovers network topology using the IGP database with the collection of node properties and interface and port discovery using SNMP. This is typically the first NIMO that is configured before other NIMOs, because it provides the basic data collection needed. This NIMO provides full topology discovery and, although not common, topology discovery without interfaces or port detail collection. The network model resulting from this topology discovery is used as the source network for additional collections. It provides the core node, circuit, and interface information used by other NIMOs.

Note

- It is assumed that you are in the middle of creating a network model when performing the tasks described in this topic. For more information, see Create a Network Model.
- Although this topic shows how to use the Expert Mode for configuration, it can be referred to for configuring options using the WAE UI or WAE CLI.

Before you begin

Device and network access profiles must be configured. See Configure Network Access.

Step 1 Choose **topo-igp-nimo** as the NIMO type.

Step 2 Choose the network access type.

Step 3 Enter the management IP address of the seed router.

Step 4 Choose the IGP protocol that is running on the network.

Step 5 From the collect-interfaces field, choose **true** to discover the full network topology.

Step 6 (Optional) To exclude individual nodes from collection, click the **node-blacklist** tab and enter the applicable IP addresses of the nodes.

Note For more information on advanced options, see IGP Topology Advanced Options, on page 5. From the WAE UI, you can also hover your mouse over each field to view tooltips.

Step 7 Click the **Commit** button.

Step 8 Click **run-collection** > **Invoke run-collection**.

Step 9 To verify that the collection ran successfully, navigate to back to the network (**/wae:networks/network/<*network-name*>**) and click the **model** tab.

Step 10 Click **nodes**. A list of nodes and details appears, indicating a successful collection.

What to do next

Use this network model as the source network to configure additional collections. See NIMO Descriptions, on page 1.

IGP Topology Advanced Options

This topic describes advanced options available when running IGP topology collection.

Option	Description
igp	
backup-router	Secondary seed router to use for automatic failover.
get-segments	Collect segment routing data from the IS-IS database. Only valid for IS-IS on Cisco IOS XR routers.
ospf-area	Collect a single OSPF area, or collect all areas. The area ID can be specified as an integer or as an IP address. If set to 'all', ABRs are identified from area 0 information and logged into for non-zero area information.
ospf-proc-id	OSPF process ID to use when there are multiple OSPF processes. The value is a positive integer.
database-file	File in which to write the raw IGP database.
offline	Run IGP discovery in offline mode. In offline mode, no login access to routers is performed; instead, the necessary configs must be provided in the database file. Offline mode is mainly used for testing.
login-record-mode	Record the discovery process. If set to 'record', messages to and from the live network are recorded in the login-record-dir as the tool runs. Used for debugging.
login-record-dir	Directory in which to save the login record. Used for debugging.
nodes	
remove-node-suffix	Remove node suffixes from node names if the node contains this suffix. For example, 'company.net' removes the domain name for the network.
nodes | interfaces	
net-recorder	If set to 'record', SNMP messages to and from the live network are recorded in the net-record-file as discovery runs. Used for debugging.
net-record-file	Directory in which to save the SNMP record. Used for debugging.
interfaces	
find-parallel-links	Find parallel links that aren't in the IGP database (when IS-IS TE extensions aren't enabled).

Option	Description
ip-guessing	Level of IP address guessing to perform for interfaces that are not present in the topology database. (Used when IS-IS TE extensions aren't enabled.) • off—Perform no guessing. • safe—Choose guesses that have no ambiguity. • full—Make best-guess decisions when there is ambiguity.
lag	Enable LAG discovery of port members.
lag-port-match	Determine how to match local and remote ports in port circuits. • exact—Match based on LACP. • none—Do not create port circuits. • guess—Create port circuits to match as many ports as possible. • complete—Match based on LACP first, and then try to match as many as possible.
cleanup-circuits	Remove circuits that don't have IP addresses associated to interfaces. Circuit removal is sometimes required with IS-IS databases to fix IS-IS advertising inconsistencies.
copy-descriptions	Copy physical interface descriptions to logical interfaces if there is only one logical interface and its description is blank.
get-physical-ports	Collect L3 physical ports for Cisco. Collect physical ports if there is an L1 connection underneath.
min-prefix-length	Minimum prefix length to allow when finding parallel links. All interfaces with equal or larger prefix lengths (but less than 32) are considered.
min-guess-prefix-length	Minimum IP guessing prefix length. All interfaces with equal or larger prefix lengths are considered.

BGP-LS Topology Collection Using XTC

BGP-LS XTC topology (topo-bgpls-xtc-nimo) discovers Layer 3 topology using BGP-LS via XTC. It uses raw XTC data as the source for topology. Node and interface/port properties are discovered using SNMP. For testing purposes, you can also use BGP-LS XTC topology discovery using XTC only (extended topology discovery disabled) when no SNMP access is available. The network model resulting from topology discovery is used as the source network for additional collections because it provides the core node/circuit/interface information used by other NIMOs.

BGP-LS XTC topology discovery ***using XTC only*** is used as a source for only some NIMOs because it does not collect the necessary information needed by most NIMOs.

Before you begin

- Device access and network access must be configured. For more information, see Configure Device Access Using the Expert Mode and Configure Network Access.

- An XTC agent must be configured and running. For more information, see Configuring XTC Agents Using the Expert Mode.

Step 1 From the Expert Mode, navigate to **/wae:networks**.

Step 2 Click the plus (+) sign and enter a network model name. We recommend a unique name that contains the NIMO name; for example, networkABC_bgpls_xtc.

Step 3 Click **Add**.

Step 4 Click the **nimo** tab.

Step 5 From the **Choice - nimo-type** drop-down list, choose **topo-bgpls-xtc-nimo**.

Step 6 Enter the following information:

- **network-access**—Choose the network access.
- **xtc-host**—Choose an XTC agent.
- **backup-xtc-host**—Choose a backup XTC agent. You can enter the same XTC agent if you do not have a backup.
- **asn**—Enter 0 to collect information from all autonomous systems in the network, or enter the autonomous system number (ASN) to collect information only from a particular ASN. For example, if the XTC agent has visibility to ASN 64010 and ASN 64020, enter 64020 to collect information only from ASN 64020. You must enter an ASN if you plan to use the as-merger NIMO to consolidate different AS models into one network model.
- **igp-protocol**—Choose the IGP protocol that is running on the network.
- **extended-topology-discovery**—Choose **true** to discover the full network topology (node and interfaces).

Note For more information on advanced options, see BGP-LS XTC Advanced Options, on page 8. From the WAE UI, you can also hover your mouse over each field to view tooltips.

Step 7 (Optional) To exclude individual nodes from collection, click the **node-blacklist** tab and enter the applicable IP addresses of the nodes. For example, you might not want to see the XTC nodes in the collection.

Step 8 Click the **Commit** button.

Step 9 Click **run-xtc-collection** > **Invoke run-collection**.

Step 10 To verify that the collection ran successfully, navigate to back to the network (**/wae:networks/network/<*network-name*>**) and click the **model** tab.

Step 11 Click **nodes**. A list of nodes and details appears, indicating a successful collection.

Example

For example, if using the WAE CLI (in config mode), enter:

```
# networks network <network-model-name> nimo topo-bgpls-xtc-nimo network-access
<network-access-ID>
# networks network <network-model-name> nimo topo-bgpls-xtc-nimo xtc-host <XTC-agent>
# networks network <network-model-name> nimo topo-bgpls-xtc-nimo backup-xtc-host
<XTC-agent-backup>
# networks network <network-model-name> nimo topo-bgpls-xtc-nimo asn <ASN-number>
# networks network <network-model-name> nimo topo-bgpls-xtc-nimo igp-protocol
<IGP-protocol-type>
# networks network <network-model-name> nimo topo-bgpls-xtc-nimo extended-topology-discovery
 <true-or-false>
```

What to do next

After performing this task, you can use this network model as the source network to configure additional collections. For more information, see NIMO Descriptions, on page 1.

BGP-LS XTC Advanced Options

This topic describes advanced options available when running BGP-LS topology collection using XTC.

Option	Description
nodes	
remove-node-suffix	Remove node suffixes from node names if the node contains this suffix. For example, 'company.net' removes the domain name for the network.
nodes | interfaces	
net-recorder	If set to 'record', SNMP messages to and from the live network are recorded in the net-record-file as discovery runs. Used for debugging.
net-record-file	Directory in which to save the SNMP record. Used for debugging.
interfaces	
find-parallel-links	Find parallel links that aren't in the IGP database (when IS-IS TE extensions aren't enabled).
ip-guessing	Level of IP address guessing to perform for interfaces that are not present in the topology database. (Used when IS-IS TE extensions aren't enabled.) • off—Perform no guessing. • safe—Choose guesses that have no ambiguity. • full—Make best-guess decisions when there is ambiguity.
lag	Enable LAG discovery of port members.
lag-port-match	Determine how to match local and remote ports in port circuits. • exact—Match based on LACP. • none—Do not create port circuits. • guess—Create port circuits to match as many ports as possible. • complete—Match based on LACP first, and then try to match as many as possible.
cleanup-circuits	Remove circuits that don't have IP addresses associated to interfaces. Circuit removal is sometimes required with IS-IS databases to fix IS-IS advertising inconsistencies.
copy-descriptions	Copy physical interface descriptions to logical interfaces if there is only one logical interface and its description is blank.
get-physical-ports	Collect L3 physical ports for Cisco. Collect physical ports if there is an L1 connection underneath.

Option	Description
min-prefix-length	Minimum prefix length to allow when finding parallel links. All interfaces with equal or larger prefix lengths (but less than 32) are considered.
min-guess-prefix-length	Minimum IP guessing prefix length. All interfaces with equal or larger prefix lengths are considered.

NIMO Collection Consolidation

The aggregator uses the Delta Aggregation Rules Engine (DARE) to combine user-specified NIMOs into a single consolidated network model. The aggregator reads the capabilities of source NIMOs. For more information on aggregator functions, see Network Models.

Note For networks using XTC, you can get automated network updates to obtain real-time network models that can be used for automation applications. For more information, see Automation Applications.

Before you begin

Configure NIMOs that you want to include in the final network model. It is important not to run a collection or execute these NIMOs until after the initial aggregator configuration.

Step 1 Create an empty network. This will be the final consolidated network model. From the Expert Mode, navigate to **/wae:networks**, click the plus (+) sign, and enter a final network model name.

Step 2 Navigate to **/wae:wae/components/dare:aggregators** and select the aggregator tab.

Step 3 Click the plus (+) sign.

Step 4 From the drop-down destination list, select the final network and click **Add**.

Step 5 Click the source link.

Step 6 Click the plus (+) sign to add source NIMOs. Repeat until all source NIMOs you want to consolidate collections for under the final network model are added.

Step 7 Click **OK**.

Step 8 (Optional) Rules are automatically generated when a source is added or removed from the source list, or when a refresh is invoked in the resulting aggregator (dependent on the capabilities reported by source NIMOs). To select a rule set other than the default, navigate back to **/wae:wae/components/dare:aggregators/aggregator/<*network_name*>/aggregator** and select an option under the **rule-set** drop-down list. To edit rule sets, navigate to **/wae:wae/components/dare:rule-sets/rule-set** and select either default or full.

Step 9 Click the **Commit** button.

Step 10 Run the source NIMOs. The final network model will update with the latest information from the source network models. See also Aggregator and Multi-Layer Collection CLI Configuration Example, on page 10.

Example

If using the WAE CLI (in config mode), enter:

```
# wae components aggregators aggregator <final-network-model>
# sources source <nimo_1>
# sources source <nimo_2>
# commit
```

After the aggregator is configured, then run the source NIMOs.

Aggregator and Multi-Layer Collection CLI Configuration Example

This example shows how to configure the aggregator to combine Layer 3 and Layer 1 network model information using the CLI.

The following shows that L1 (optical) and L3 (topo-igp-nimo) network models have been configured on the network. For more information on how to configure the optical NIMO and topo-igp-nimo, see IGP Topology Collection, on page 4 and Configure Multi-Layer Collection.

```
# show running-config networks network nimo
```

Layer 1 network model:

```
networks network l1-network
    nimo optical-nimo source-network l3-network
    nimo optical-nimo network-access cisco:access
    nimo optical-nimo optical-agents cisco:network
        advanced use-configure-l3-l1-mapping true
        advanced l3-l1-mapping    bg1_mapping
    !
```

Layer 3 network model:

```
    nimo topo-igp-nimo network-access cisco:access
    nimo topo-igp-nimo seed-router 10.225.121.60
    nimo topo-igp-nimo igp-protocol isis
    nimo topo-igp-nimo collect-interfaces true
    nimo topo-igp-nimo node-blacklist 10.11.255.12
    !
    nimo topo-igp-nimo advanced interfaces lag true
```

Note Collection has not yet been done on the configured L1 and L3 network models.

Configure the aggregator.

```
# config
# wae components aggregators aggregator l1-l3-final-model
# sources source l1-network
# sources source l3-network
```

```
# commit
```

After the aggregator is configured, run the L3 and L1 collections.

```
# networks network l3-network nimo topo-igp-nimo run-collection
```

A status message will appear once collection is complete. Once it completes, check to see that the nodes are populated.

```
# show running-config networks network l3-network model nodes node
```

You can also check to see that the final model is also populated with L3 information.

```
# show running-config networks network l1-l3-final-model model nodes node
```

Run the L1 network collection.

```
# networks network l1-network nimo optical-nimo build-optical-topology
```

You can check again to see that the final model is now populated with L1 information.

```
# show running-config networks network l1-l3-final-model model nodes node
```

You can also open WAE Design to view the final network model (**File** > **Open from** > **WAE Automation Server** and select the final network model.

Autonomous System (AS) Model Consolidation

The as-merger NIMO resolves interfaces, circuits, etc., that are shared between AS models to create a single, consolidated network model.

Before you begin

- Confirm that collection has been completed on the individual AS network models that you want to merge.

Note BGP collection must be part of all AS models.

- Any AS network models that use the topo-bgpls-xtc NIMO must each have an Autonomous System Number (ASN) assigned to it. For more information, see BGP-LS Topology Collection Using XTC, on page 6.

Step 1 From the Expert Mode, navigate to **/wae:networks**.

Step 2 Click the plus (+) sign and enter a name for the consolidated AS network model.

Step 3 Click **Add**.

Step 4 Click the **nimo** tab.

Step 5 From the **Choice - nimo-type** drop-down list, choose **as-merger-nimo**.

Step 6 Click **as-merger-nimo**.

Step 7 From **sources**, add the individual AS models you want to merge.

Step 8 From **subtrees**, enter **model** as the value.

Step 9 From **generate capabilities**, select one of the following:

- **false**—Select this option if the AS sources are DARE networks.
- **true**—Select this option if the AS sources are not DARE networks and are created by other NIMOs.

Step 10 Click the **Commit** button.

Step 11 Click **merge**.

Example

If using the WAE CLI (in config mode), enter:

```
# config
# networks network <as-merger-name> nimo as-merger sources [ <AS1-source> <AS2-sourse>
<ASn-source> ]
# networks network <as-merger-name> nimo as-merger subtrees [ model ]
# networks network <as-merger-name> nimo as-merger generate-capabilities <flag>
# commit
```

For example:

```
# config
# networks network AS100AS200 nimo as-merger sources [ AS6100 AS6200 ]
# networks network AS100AS200 nimo as-merger subtrees [ model ]
# networks network AS100AS200 nimo as-merger generate-capabilities <false>
# commit
```

VPN Collection

The VPN Collection (topo-vpn-nimo) discovers Layer 2 and Layer 3 VPN topology.

Before you begin

Network topology collection must be complete. For more information, see Create a Network Model.

Step 1 From the Expert Mode, navigate to **/wae:networks**.

Step 2 Click the plus (+) sign and enter a network model name. We recommend a unique name that contains the source network and NIMO names; for example, networkABC_vpn.

Step 3 Click **Add**.

Step 4 Click the **nimo** tab.

Step 5 From the **Choice - nimo-type** drop-down list, choose **topo-vpn-nimo**.

Step 6 Click **topo-vpn-nimo** and enter the following:

- **source-network**—Choose the applicable network model that contains basic topology information.
- **network-access**—Choose the network access.

Step 7 Click the **vpn-types** tab.

Step 8 Click the plus (+) icon to add at least one VPN type:

- **VPWS**—Add this type when Virtual Private Wire Service is being used in the network.
- **L3VPN**—Add this type when Layer 3 VPN is being used in the network.

Step 9 Click the **Commit** button.

Step 10 Navigate back to the **topo-vpn-nimo** tab and click **run-collection** > **Invoke run-collection**.

LSP Collection Using NSO NEDs

The lsp-config-nimo discovers LSP configuration using NEDs. This collection lets you deploy changes to LSPs, named paths, and segment lists in the network. For a configuration example of creating segment lists and LSP paths, see Create Segment Routing LSPs, on page 14.

Before you begin

- You must have the Network Services Orchestrator (NSO) installed on your system and vendor network element drivers (NEDs). Contact your Cisco representative to acquire the appropriate NEDs.
- Confirm that a WAE LSA configuration with an NSO instance is configured. See Install LSA Packages for more information.
- A basic topology network model must exist.

Step 1 Copy the appropriate LSP NIMO configuration files from the WAE installation directory to your WAE run-time packages directory.

a) From the Unix shell, navigate to ***<wae_installation_directory>*/packages/cisco-wae-lsp-config-nimo** directory.

b) Copy the appropriate LSP NIMO package (for example, **cisco-wae-lsp-config-nimo-rfs**) directory to <*wae_run_time_directory*>/packages.

Step 2 Copy the appropriate NED packages to <*wae_run_time_directory*>/packages.

Step 3 From the Expert Mode, navigate to **/wae:networks**.

Step 4 Click the plus (+) sign and enter a network model name. We recommend a unique name that contains the source network and NIMO names; for example, networkABC_lsp_config.

Step 5 Click **Add**.

Step 6 Click the **nimo** tab.

Step 7 From the **Choice - nimo-type** drop-down list, choose **lsp-config-nimo**.

Step 8 Click **lsp-config-nimo** and enter the source topology network.

Note
- The lsp-config-nimo must follow a topology NIMO. For example, you cannot choose a layout network (layout-nimo) as the source network.
- For information on advanced options, see LSP Configuration Advanced Options, on page 14. From the WAE UI, you can also hover your mouse over each field to view tooltips.

Step 9 Click the **Commit** button.

Step 10 Click **run-collection** > **Invoke run-collection**.

LSP Configuration Advanced Options

This topic describes advanced options available when configuring the lsp-config-nimo.

Option	Description
Attributes	
action-timeout	Timeout (in minutes) to use for all lsp-config-nimo actions. The default value is 0. Large networks might require a larger value.
create-nodes	Whether the device-sync action creates nodes. The default value is 'false'.
perform-sync-from	By default, the run-collection and device-sync actions first perform a device sync-from, which takes several seconds per device. For large networks, it is more efficient to set this value to 'false' and perform a single device sync-from before performing a run-collection. The default value is 'false'.
preserve-device-config	Whether the run-collection commits changes to the device model and how 're-deploy reconcile' is performed. If set to 'true', collection will also preserve portions of the device model not represented in the service model since reconciliation is done with the 'keep-non-service-config' option. The default option is 'true'.
Actions	
copy-topology	Copy only the source network to this network. **Warning** This action removes all LSPs.
device-sync	Discover and reconcile LSPs in the topology network. The source network is not copied.
reconcile	Reconcile the LSPs in the topology network with the device model.

Create Segment Routing LSPs

The following procedure describes the commands to create segment routing LSPs.

Step 1 Launch the WAE CLI from the WAE run-time directory and enter configuration mode.

```
waerun# wae_cli -C
wae@wae# config
wae@wae$#
```

Step 2 Create a segment list with two hops.

```
wae@wae(config)# networks network <network_name> model nodes node <node_name>
 segment-lists segment-list <segment_list_name>
```

The first hop is a node hop.

```
wae@wae(config-segment-list-<segment_list_name>)# hops hop 1 node node-name <node_name>
wae@wae(config-hop-1)# exit
```

The second hop is an interface hop.

```
wae@wae(config-segment-list-<segment_list_name>)# hops hop 2 interface node-name <node_name>
interface-name <interface_name>
wae@wae(config-hop-2)# exit
wae@wae(config-segment-list-<segment_list_name>)# exit
```

Step 3 Create an LSP with two LSP paths.

```
wae@wae(config)# networks network <network_name> model nodes node <node_name>
 segment-lists segment-list <segment_list_name>
```

The first path is a PCE-delegated path.

```
wae@wae(config-node-<node_name>)# lsps lsp <lsp_name> destination <destination> lsp-paths lsp-path
1 type segment-routing pce-delegated true
wae@wae(config-lsp-path-1)# exit
```

The second path uses the previous segment list that was created earlier.

```
wae@wae(config-lsp-<lsp_name>)# lsp-paths lsp-path 2 type segment-routing segment-list <segment_list>
wae@wae(config)# commit
```

Example

For example:

```
wae@wae(config)# networks network <network_name> model nodes node <node_name> segment-lists
 segment-list <segment_list_name>
wae@wae(config-segment-list-<segment_list_name>)# hops hop 1 node node-name <node_name>
wae@wae(config-hop-1)# exit
wae@wae(config-segment-list-<segment_list_name>)# hops hop 2 interface node-name <node_name>
 interface-name <interface_name>
wae@wae(config-hop-2)# exit
wae@wae(config-segment-list-<segment_list_name>)# exit
wae@wae(config-node-<node_name>)# lsps lsp <lsp_name> destination <destination> lsp-paths
lsp-path 1 type segment-routing pce-delegated true
wae@wae(config-lsp-path-1)# exit
wae@wae(config-lsp-<lsp_name>)# lsp-paths lsp-path 2 type segment-routing segment-list
<segment_list>
wae@wae(config)# commit
```

PCEP LSP Collection Using XTC

PCEP LSP discovery using XTC (lsp-pcep-xtc-nimo) uses the data collected from the bgpls-xtc-nimo and appends PCEP LSP information, thus creating a new network model.

Before you begin

Confirm that BGP-LS topology collection using XTC (bgpls-xtc-nimo) has been completed for a network. You will need to use this model as the source network for collecting PCEP LSPs. For more information, see BGP-LS Topology Collection Using XTC, on page 6.

Step 1 From the Expert Mode, navigate to **/wae:networks**.

Step 2 Click the plus (+) sign and enter a network model name. We recommend a unique name that contains the source network and NIMO names; for example, **`networkABC_lsp_pcep_xtc`**.

Step 3 Click **Add**.

Step 4 Click the **nimo** tab.

Step 5 From the **Choice - nimo-type** drop-down list, choose **lsp-pcep-xtc-nimo**.

Step 6 Click **lsp-pcep-xtc-nimo** and enter the source network. This is the network model that contains topology information collected using the bgpls-xtc-nimo.

Step 7 Click the **xtc-hosts** tab.

Step 8 Click the plus (+) icon and enter the following:

- **name**—Enter an XTC hostname. This can be any arbitrary name.
- **xtc-host**—From the drop-down list, choose one of the XTC hosts that was previously configured. For more information, see Configuring XTC Agents Using the Expert Mode.

Step 9 Click the **Commit** button.

Step 10 Click **run-xtc-collection** > **Invoke run-collection**.

Step 11 To verify that the collection ran successfully, navigate to back to the network (**/wae:networks/network/<*network-name*>**) and click the **model** tab.

Step 12 Click **nodes**. A list of nodes and details appears, indicating a successful collection.

Step 13 Choose one of the nodes that you know has an LSP and click the **lsps** tab.

Step 14 Click the **lsp** link. A table with a list of discovered LSPs appears.

LAG Ports and LMP Interface Collection

The port-cfg-parse NIMO discovers LAG ports and Link Management Protocol (LMP) interfaces from router configurations in the network. This NIMO is used for multi-layer collection.

Note You cannot use the WAE UI to configure this collection.

Before you begin

- A topology network model must exist. See Create a Network Model.
- The Configuration Parsing agent must be configured and running. For more information, see Configuring the Configuration Parsing Agent.

Step 1 From the Expert Mode, navigate to **/wae:networks**.

Step 2 Click the plus (+) sign and enter a network model name. We recommend a unique name that contains the source network and NIMO names; for example, networkABC_port-cfg-parse.

Step 3 Click **Add**.

Step 4 Click the **nimo** tab.

Step 5 Choose **port-cfg-parse-nimo** as the NIMO type.

Step 6 Click **port-cfg-parse-nimo** and enter the following information:

- **source-network**—Choose the applicable network model that contains topology information.
- **cfg-parse-agent**—Choose a configuration parsing agent.

Note For information on advanced options, see Port Config Parse Advanced Options, on page 17.

Step 7 Click the **Commit** button.

Step 8 Click **run-collection** > **Invoke run-collection**.

What to do next

After performing this task, you can use this network model as a source network to configure additional collections. For more information, see NIMO Descriptions, on page 1.

Port Config Parse Advanced Options

This topic describes advanced options available when creating the port-cfg-parse NIMO.

Option	Description
lag	Enable LAG discovery of port members.
lmp	Enable discovery of LMP interfaces.

BGP Peer Collection

The topo-bgp-nimo discovers BGP topology via SNMP and login. It uses a topology network (typically an IGP topology collection model) as its source network and adds BGP links to external ASN nodes.

Before you begin

A topology network model must exist. See Create a Network Model.

Step 1 From the Expert Mode, navigate to **/wae:networks**.

Step 2 Click the plus (+) sign and enter a network model name. We recommend a unique name that contains the source network and NIMO names; for example, networkABC_topo_bgp.

Step 3 Click **Add**.

Step 4 Click the **nimo** tab.

Step 5 From the **Choice - nimo-type** drop-down list, choose **topo-bgp-nimo**.

Step 6 Click **topo-bgp-nimo** and enter the following information:

- **source-network**—Choose the applicable network model that contains basic topology information.
- **network-access**—Choose a network access profile that was previously configured.
- **min-prefix-length**—(Optional) Enter the min-prefix-length to control how restrictive IPv4 subnet matching is in discovering interfaces as BGP links.
- **min-IPv6-prefix-length**—(Optional) Enter the min-IPv6-prefix-length to control how restrictive IPv6 subnet matching is in discovering interfaces as BGP links.
- **login-multi-hop**—(Optional) Choose whether to disable login-multihop if you do not want to log in to routers that potentially contain multihop peers.

For more information on advanced options, see BGP Topology Advanced Options, on page 18.

Step 7 Click the **peer-protocol** tab and enter applicable IPv4 and IPv6 addresses.

Step 8 Click the **Commit** button.

Step 9 Click **run-collection** > **Invoke run-collection**.

BGP Topology Advanced Options

This topic describes advanced options available when running BGP topology collection.

Option	Description
force-login-platform	Override platform detect and use the specified platform. Valid values: cisco, juniper, alu, huawei.
fallback-login-platform	Fallback vendor in case platform detection fails. Valid values: cisco, juniper, alu, huawei.
try-send-enable	When logging in to a router, send an enable password if the platform type is not detected. This action has the same behavior as '-fallback-login-platform cisco'.
internal-asns	Specify internal ASNs. If used, the specified ASNs are set to internal; all others are set to external. The default is to use what is discovered.
asn-include	Specify ASNs of interest. If used, peer discovery is restricted to this list. The default is to peer with all discovered external ASNs.
find-internal-asn-links	Find links between two or more internal ASNs. Normally this action is not required because IGP discovers these links.

Option	Description
find-non-ip-exit-interface	Search for exit interfaces that are not represented as next-hop IP addresses, but rather as interfaces (which are rare). **Note** This action increases the amount of SNMP requests for BGP discovery, which affects performance.
find-internal-exit-interfaces	Collect exit interfaces to internal ASNs.
get-mac-address	Collect source MAC addresses of BGP peers connected to an Internet Exchange public peering switch. This action is required only for MAC accounting.
use-dns	Whether to use DNS to resolve BGP IP addresses.
force-check-all	Check all routers even if there is no indication of potential multi-hop peers. This action could be slow.
net-recorder	If set to 'record', SNMP messages to and from the live network are recorded in the net-record-file as discovery runs. Used for debugging.
net-record-file	Directory in which to save the SNMP record. Used for debugging.
login-record-mode	Record the discovery process. If set to 'record', messages to and from the live network are recorded in the login-record-dir as the tool runs. Used for debugging.
login-record-dir	Directory in which to save the login record. Used for debugging.

LSP Collection Using SNMP

The lsp-snmp-nimo discovers LSP information using SNMP.

Before you begin

A basic topology network model must exist. See Basic Topology Collection, on page 4.

Step 1 From the Expert Mode, navigate to **/wae:networks**.

Step 2 Click the plus (+) sign and enter a network model name. We recommend a unique name that contains the source network and NIMO names; for example, networkABC_lsp_config.

Step 3 Click **Add**.

Step 4 Click the **nimo** tab.

Step 5 From the **Choice - nimo-type** drop-down list, choose **lsp-snmp-nimo**.

Step 6 Click **lsp-snmp-nimo** and enter the following:

- **source-network**—Choose the applicable network model that contains basic topology information.
- **network-access**—Choose a network access profile that was previously configured.
- **get-frr-lsps**—Choose **true** if you want to discover Multiprotocol Label Switching (MPLS) Fast Reroute (FRR) LSP (backup and bypass) information.

Step 7 Click the **Commit** button.

Step 8 Click **run-collection** > **Invoke run-collection**.

Segment Routing LSP Traffic Collection

Segment Routing (SR) LSP Traffic Collection (sr-traffic-matrix-nimo) discovers SR LSP traffic. This NIMO enables the generation of demands between external interfaces of a network from collected telemetry data.

Before you begin

- A basic topology network model must exist. See IGP Topology Collection, on page 4 or BGP-LS Topology Collection Using XTC, on page 6.
- Telemetry must be configured on the router.

Note You cannot use the WAE UI to configure this collection.

Step 1 From the Expert Mode, navigate to **/wae:networks**.

Step 2 Click the plus (+) sign and enter a network model name. We recommend a unique name that contains the source network and NIMO names; for example, networkABC_sr_traffic_matrix.

Step 3 Click **Add**.

Step 4 Click the **nimo** tab.

Step 5 From the **Choice - nimo-type** drop-down list, choose **sr-traffic-matrix-nimo**.

Step 6 Click **sr-traffic-matrix-nimo** and enter the source network.

Step 7 Click the **Commit** button.

Step 8 Click **run-collection** > **Invoke run-collection**.

Example

If using the WAE CLI (in config mode), enter:

```
# networks network <network-model-name> nimo sr-traffic-matrix-nimo source-network
<source-network>

# commit
```

Continuous Collection

The traffic-poll-nimo collects traffic statistics (interface measurements) using SNMP polling.

Before you begin

This NIMO requires the following:

- Basic topology network model.
- If collecting VPN traffic, a VPN network model must exist. See VPN Collection, on page 12.
- If collecting LSP traffic, an LSP network model must exist. See LSP Collection Using SNMP, on page 19.

Limitations

- Node traffic information from external interfaces is not collected.

Step 1 From the Expert Mode, navigate to **/wae:networks**.

Step 2 Click the plus (+) sign and enter a network model name. We recommend a unique name that contains the source network and NIMO names; for example, networkABC_traffic_polling.

Step 3 Click **Add**.

Step 4 Click the **nimo** tab.

Step 5 From the **Choice - nimo-type** drop-down list, choose **traffic-poll-nimo**.

Step 6 Click **traffic-poll-nimo** and enter the following:

- **source-network**—Choose the applicable network model.
- **network-access**—Choose the network access.

Step 7 To run continuous traffic collection for interfaces, click the **iface-traffic-poller** tab and enter the following:

- **enabled**—Set to **true**.
- **period**—Enter the polling period, in seconds. We recommend starting with 60 seconds. See Tuning Traffic Polling, on page 22 to tune the polling period.
- **qos-enabled**—Set to **true** if you want to enable queues traffic collection.
- **vpn-enabled**—Set to **true** if you want to enable VPN traffic collection. If set to true, confirm that the source network model has VPNs enabled.

Step 8 To run continuous traffic collection for LSPs, click the **lsp-traffic-poller** tab and enter the following:

- **enabled**—Set to **true**.
- **period**—Enter the polling period, in seconds. We recommend starting with 60 seconds. See Tuning Traffic Polling, on page 22 to tune the polling period.

Step 9 Click the **Commit** button.

Step 10 Navigate back to the **traffic-poll-nimo** tab and click **run-snmp-traffic-poller** > **Invoke run-snmp-poller**. To stop continuous collection in the future, click **stop-snmp-traffic-poller**.

Traffic Polling Advanced Options

This topic describes advanced options available when configuring continuous collection (traffic-poll-nimo).

Option	Description
snmp-traffic-poller	
net-recorder	This option is typically used for debugging. Set to **record** to record SNMP messages to and from the live network in the net-record-file when discovery is running.
net-record-file	Enter the filename where recorded SNMP messages are saved.
verbosity	Set the poller logging level. The default is 40. • 40—INFO • 50—DEBUG • 60—TRACE
stats-computing-minimum-window-length	Enter the minimum window length for traffic calculation, in seconds. The default is 300 seconds.
stats-computing-maximum-window-length	Enter the maximum window length for traffic calculation, in seconds. The default is 450 seconds.
raw-counter-ttl	Enter how long to keep raw counters, in minutes. The default is 15 minutes.
snmp-traffic-population	
scheduler-interval	Enter the interval to perform traffic population, in seconds. The default is 300 seconds. It will send traffic statistics to the configuration database (CDB). If set to 0 (typically set when using the Bandwidth on Demand application), WMD will pull the traffic statistics from the RPC API. The traffic statistics will not be sent to CDB.
connect-timeout	Enter the maximum execution time for traffic population, in minutes.

Tuning Traffic Polling

To run traffic polling efficiently, do the following:

1. Start with the default options and run continuous collection for several hours. The default values are:

```
iface-traffic-poller/period = 60
lsp-traffic-poller/period = 60
advanced/snmp-traffic-poller/stats-computing-minimum-window-length = 300
advanced/snmp-traffic-poller/stats-computing-maximum-window-length = 450
advanced/snmp-traffic-poller/raw-counter-ttl = 15
advanced/snmp-traffic-population/scheduler-interval = 300
```

2. View the poller.log file. By default, the file is located in *<wae_run_time_directory>*/logs/*<network_name>*-poller.log.
3. Search for and filter the following text:
   - Interface Traffic Poller: Collection complete. Duration:
   - LSP Traffic Poller: Collection complete. Duration:

4. Note the duration on average and during a worst-case scenario.

For example, interface polling takes 30 seconds on average and 45 seconds in a worst-case scenario. LSP polling takes 90 seconds on average and a maximum of 120 seconds. To run interface polling efficiently, we recommend starting with the worst-case (higher) numbers. In this example, start with the interface poller every 45 seconds and the LSP poller every 120 seconds. Then, as time passes, you can tune these numbers as needed.

To calculate traffic, you need at least two counters (two polling runs). Counters are chosen among those stored in memory using a sliding window. The advanced `raw-counter-ttl` option dictates how long the counters are kept. The advanced `stats-computing-minimum-window-length` and `stats-computing-maximum-window-length` options specify the size of the sliding window. The basic calculation on how to configure these parameters is:

```
stats-computing-minimum-window-length = ( poller duration ) * 5
stats-computing-maximum-window-length = stats-computing-minimum-window-length * 1.5
raw-counter-ttl = stats-computing-minimum-window-length * 3 / 60
```

In this example, because LSP collection takes longer than interface collection, use the LSP worst-case duration (120 seconds) to determine the sliding window size. You get the following numbers:

```
stats-computing-minimum-window-length = 120*5 = 600
stats-computing-maximum-window-length = stats-computing-minimum-window-length * 1.5 = 600
* 1.5 = 900
raw-counter-ttl = stats-computing-minimum-window-length * 3 / 60 = 30
```

You can run traffic calculation as often as every 30 to 45 seconds, or as slowly as every 120 seconds, as configured with the advanced `scheduler-interval` option. Set it to 120 seconds to conserve CPU. It might take a while to calculate traffic in large networks, so make sure to set an adequate **`connect-timeout`** option. You can find the actual period in `logs/ncs-java-vm.log`. For example:

```
Traffic calculation took (ms) 3750
```

You can tune these parameters more aggressively or conservatively. We recommend starting with more conservative settings, and then tuning them as needed. If you see traffic being dropped from the output, you can increase the `stats-computing-maximum-window-length` and `raw-counter-ttl` options accordingly.

Network Model Visualization

The layout-nimo adds layout properties to a source network model to improve visualization when importing the plan file into WAE Design. The NIMO automatically records changes to the layout properties. When the source network model changes, the layout of the destination model is updated.

The layout in the destination network serves as a template that is applied to the source network. The resulting network is saved as the new destination network. If the source layout contains no layout information, the layout from the destination network is simply added to the source network. If the source network contains layout information, that layout is maintained unless there is a conflict with the layout in the destination network. If a conflict exists, the layout information in the destination network takes precedence over the information in the source network.

For example, assume that a new L1 node is added to the source network with a corresponding site assignment. This L1 node is then added to the destination network with its site assignment. Now assume that an existing L1 node has a different site assignment in the source and destination networks. In this case, the site assignment in the destination network is retained.

There are two steps:

1. Create a new network model using the layout-nimo.
2. Add a layout template to the new network model using WAE Design and then send a patch. For more information, see the Cisco WAE Network Visualization Guide.

Before you begin

- A basic topology network model must exist. See Basic Topology Collection, on page 4.

Note You cannot use the WAE UI to configure this collection.

Step 1 From the Expert Mode, navigate to **/wae:networks**.

Step 2 Click the plus (+) sign and enter a network model name. We recommend a unique name that contains the source network and NIMO names. This procedure uses **networkABC_layout** as an example.

Step 3 Click **Add**.

Step 4 Click the **nimo** tab.

Step 5 From the **Choice - nimo-type** drop-down list, choose **layout-nimo**.

Step 6 Click **layout-nimo** and enter a source network.

Note The source network cannot have a preexisting layout template assigned to it.

Step 7 Click the **Commit** button.

Step 8 Click **run-layout** > **Invoke run-layout**.

Step 9 Launch WAE Design and choose **File** > **Open From** > **WAE Automation Server**.

Step 10 Enter the appropriate details, choose the plan file for the network model you just created (networkABC_layout), and click **OK**.

Step 11 Edit the layout. See the "Using Layouts" chapter in the Cisco WAE Network Visualization Guide.

Step 12 Create and send the patch (**Tools** > **Patches** > **Create**). See the "Patch Files" chapter in the Cisco WAE Design User Guide.

Step 13 From the Expert Mode, navigate back to the layout-nimo network model (networkABC_layout).

Step 14 Click the **layouts** tab.

Step 15 Click **layout** to confirm that the table has been populated with layout data. The next time you open the plan file from WAE Design, the topology is displayed with the saved layout properties.

Demand Deduction

Traffic can be measured on interfaces, interface queues, and LSPs. You can use demand deduction to estimate demand traffic based on any of these measurements. For information on demand deduction, see the Cisco WAE Design User Guide. The demand-deduction-nimo runs demand deduction using measured SNMP traffic and by applying demand mesh builds of end-to-end demands (traffic matrix) to the network model.

Before you begin

A consolidated network model (aggregator NIMO) with demands (for example, networkABC_demandmesh) and interface measurements (for example, networkABC_traffic_poller) must exist.

Note You cannot use the WAE UI to configure this NIMO.

Step 1 From the Expert Mode, navigate to **/wae:networks**.

Step 2 Click the plus (+) sign and enter a network model name. We recommend a unique name that contains the source network and NIMO names; for example, networkABC_demand_deduction.

Step 3 Click **Add**.

Step 4 Click the **nimo** tab.

Step 5 From the **Choice - nimo-type** drop-down list, choose **demand-deduction-nimo**.

Step 6 Click **demand-deduction-nimo** and enter the source network. The source network is typically a consolidated network model (aggregator NIMO) that has demands and traffic poller information.

Step 7 Click the **input** tab.

- **nodes**—Use measured traffic on nodes. The default value is true.
- **interfaces**—Use measured traffic on interfaces. The default value is true.
- **lsps**—Use measured traffic on LSPs.
- **remove-zero-bw-demands**—Remove any demands with no (zero) traffic (or less than the `zero-bw-tolerance` option). The default is true.
- **zero-bw-demands-tolerance**—Enter a tolerance value (less than zero) that will be considered as zero traffic.
- **demand-upper-bound**—Enter an upper bound on the demand traffic levels. A warning is issued if the upper bound is reached. The default is 10,000 Mb/s.

Step 8 Click the **Commit** button.

Step 9 Navigate back to the **demand-deduction-nimo** tab and click **run** > **Invoke run**.

Step 10 To confirm that demand deduction succeeded, navigate to **/wae:networks/network/<*network_model*>/model/demands** and see if the traffic column was populated.

Demand Mesh Creation

Demand meshes are a time-efficient way of creating numerous demands for all or part of the network. The demandmesh-creator-nimo creates a demand mesh between a set of source and destination nodes.

Before you begin

A basic topology network model must exist. See Basic Topology Collection, on page 4.

Note You cannot use the WAE UI to configure this NIMO.

Step 1 From the Expert Mode, navigate to **/wae:networks**.

Step 2 Click the plus (+) sign and enter a network model name. We recommend a unique name that contains the source network and NIMO names; for example, networkABC_demandmesh.

Step 3 Click **Add**.

Step 4 Click the **nimo** tab.

Step 5 From the **Choice - nimo-type** drop-down list, choose **demandmesh-creator-nimo**.

Step 6 Click **demandmesh-creator-nimo** and enter the source network.

Step 7 Click the **input** tab and enter:

- **source-nodes**—Enter nodes to restrict the demand mesh source to a specified set of nodes, external asynchronous systems, or external endpoints. You can view nodes from **/wae:networks/network/<*network_model*>/model/nodes**.
- **both-directories**—Include all demands from the destination to source and also from the source to destination. The default is true.
- **service-class**—Assign a service class type to demands that are created (for example, QoS). If empty, a default class is used.
- **topology**—Enter the topology. The default is that demands are applied to all topologies.
- **choice-destination: destination-nodes**—Choose this option if you want to create demands to destinations other than what has been selected as the source.
- **choice-destination: destination-equal-source**—If set to **true**, the destination node is set to source. The default is true.

Note To get a full demand mesh, do not edit any fields and set destination-equal-source to true. This picks up all the nodes in the network and creates all possible demands (including demands to the node itself).

Step 8 Click the **Commit** button.

Step 9 Navigate back to the **demandmesh-creator-nimo** tab and click **run** > **Invoke run**.

Step 10 To confirm that demands have been created, navigate to **/wae:networks/network/<*network_model*>/model/demands**. The table is populated with demand information.

Running External Scripts Against a Network Model

The external-executable-nimo lets you run a customized script against a selected network model. You might want to do this when you want specific data from your network that existing WAE NIMOs do not provide. In this case, you take an existing model created in WAE and append information from a custom script to create a final network model that contains the data you want.

Before you begin

You must have a source network model and a custom script.

Note You cannot use the WAE UI to configure this NIMO.

Step 1 From the Expert Mode, navigate to **/wae:networks**.

Step 2 Click the plus (+) sign and enter a network model name. We recommend a unique name that is easily identifiable; for example, networkABC_my_script.

Step 3 Click the **nimo** tab.

Step 4 From the **Choice - nimo-type** drop-down list, choose **external-executable-nimo**.

Step 5 Click **external-executable-nimo** and select the source network.

Step 6 Click the **advanced** tab and enter the following:

- **input-file-version**—Enter the plan file version of the source network model, such as 6.3, 6.4, and so on. The default is 7.0.
- **input-file-format**—Specify the plan file format of the source network model. The default is .pln.
- **argv**—Enter arguments (in order) that are required for the script to run. Enter $$input for the source network model and $$output for the resulting network model (after the script runs). It is important to note that $$input, $$output, and other argv arguments must be listed in the order that is required by the script. For an example, see Running External Scripts Example, on page 27.

Step 7 From the external-executable-nimo tab, click **run...**.

Example

If using the WAE CLI (in config mode), enter:

```
networks network <network-model-name> nimo external-executable-nimo source-network
<source-network> advanced argv [ <arg_1> <arg_2> <arg_x> $$input $$output ]
admin@wae(config-network-<network-model-name>)# commit
Commit complete.
admin@wae(config-network-<network-model-name>)# exit
admin@wae(config)# exit

admin@wae# networks network <network-model-name> nimo external-executable-nimo run
```

Running External Scripts Example

This example describes how to use the external-executable-nimo with the WAE CLI. The sample python script (ext_exe_eg.py) appends a description to every interface in the network with "My IGP metric is <*value*>."

Contents of ext_exe_eg.py:

```
import sys
from com.cisco.wae.opm.network import Network

src  = sys.argv[1]
dest = sys.argv[2]

srcNet =  Network(src)

for node in srcNet.model.nodes:
    cnt = 1
    for iface in node.interfaces:
```

```
        iface.description = 'My IGP metric is ' + str(iface.igp_metric)
        cnt = cnt + 1

srcNet.write(dest)
```

In the WAE CLI, enter:

```
admin@wae(config)# networks network net_dest nimo external-executable-nimo source-network
net_src
advanced argv [ /usr/bin/python /home/user1/srcs/br1/mate/package/linux/run/ext_exe_eg.py
$$input $$output ]
admin@wae(config-network-net_dest)# commit
Commit complete.
admin@wae(config-network-net_dest)# exit
admin@wae(config)# exit

admin@wae# networks network net_dest nimo external-executable-nimo run
status true
message Changes successfully applied.
```

Confirm the script succeeded:

```
admin@wae# show running-config networks network net_dest model nodes node cr1.atl interfaces
 interface to_cr1.hst description
networks network net_dest
 model nodes node cr1.atl
  interfaces interface to_cr1.hst
   description "My IGP metric is 37"
  !
 !
!
```