Cisco WAE 6.4 Platform Configuration Guide

First Published: 2016-07-08
Last Updated: 2017-08-31

Cisco Systems, Inc.
www.cisco.com

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco website at www.cisco.com/go/offices.

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

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

© 2016–2017 Cisco Systems, Inc. All rights reserved.

CONTENTS

Platform Overview

New and Changed Information

The following table describes changes to this guide.

Date	Revision
2017-08-31	Added Chapter 5, "Collecting NetFlow Data."
2016-07-08	Initial publication

Overview

The WAN Automation Engine (WAE) platform is an open, programmable framework that interconnects software modules, communicates with the network, and provides REST and Thrift APIs to interface with external applications.

WAE Planning software provides the tools to create and maintain a model of the current network through the continual monitoring and analysis of the network and the traffic demands being placed on it. This network model contains all relevant information about a network at a given time, including topology, configuration, traffic, and routing information. You can use this information as a basis for analyzing the impact on the network due to changes in traffic demands, paths, node and link failures, network optimizations, and other possible changes to the network. With WAE Automation software, you can then deploy the results of the analyses and optimizations into the network for improved network performance or, for example, for temporary changes during network maintenance.WAE Automation uses open APIs and standardized protocols to provide interaction between applications and the network.

The WAE APIs enable you to develop applications that communicate using any language that supports REST or Thrift APIs. REST APIs are HTTP-based and are often used in the development of web-based services. Thrift is an open-source interface definition language that is used to define and create services for numerous languages.

The WAE platform has numerous use cases, including the following.

- Traffic engineering and network optimization—Add, modify, or delete network LSP configurations to improve network performance, or perform local or global optimization.
- Demand engineering—Examine the impact on network traffic flow due to adding, removing, or modifying traffic demands on the network.
- Topology and predictive analysis—Observe the impact to network performance due to changes in network topology whether driven by design or by network failures.

- Coordinated network maintenance—Explore options that minimize the impact of temporary changes to the network, such as for scheduled maintenance.
- TE tunnel programming—Examine the impact of modifying tunnel parameters, such as tunnel path and reserved bandwidth.
- CoS-aware (class of service) bandwidth on demand—Examine existing network traffic and demands, and admit a set of service-class-specific demands between routers.

Note This guide is for single-system environments only and does not contain configurations for all modules. For further configurations, such as configuring distributed environments where there is more than one instance of the WAE platform, contact your support representative.

Modules

The platform workflow (Figure 1-1) consists of ongoing data collection, network analysis and optimizations, and deployment of the resulting requirements onto the network infrastructure.

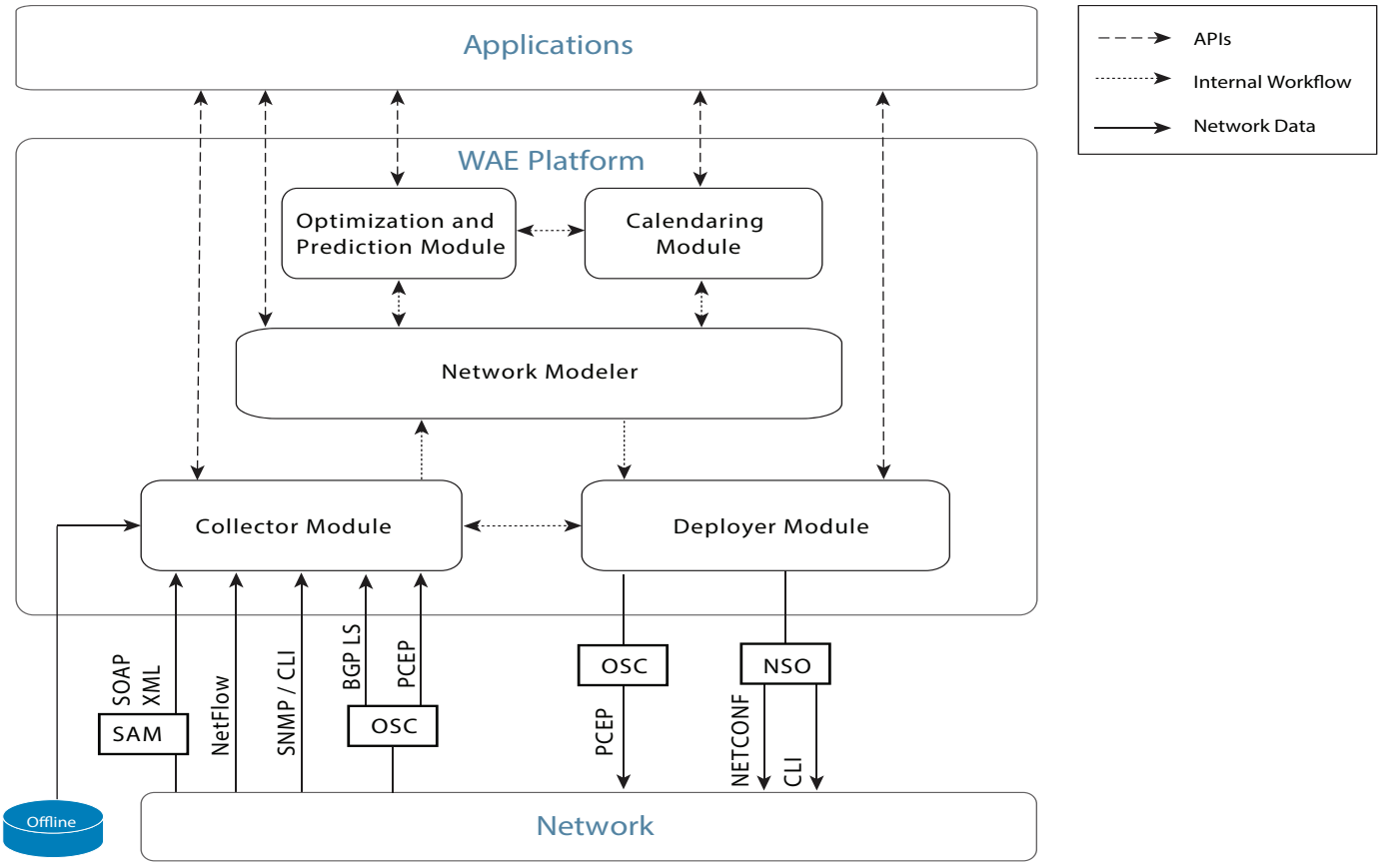
- Collector Module—Discovers the network topology, routing and peering information, and polls network for traffic, as well as other object properties. This information is exposed through APIs and is available for use by WAE Live, WAE Design, and other applications, as well as by the Network Modeler Module.
- Network Modeler Module—Maintains a current network view in the working plan area. While changes can be made to the working plan area, the recommended practice is to use staging areas. Using staging areas, multiple users can work offline simultaneously to make and test modifications before merging them with the working network model and deploying them to the network.
- Calendaring Module—Is a database used to construct a proposed network that consists of the current network model, plus any changes that are scheduled to happen between the time the network model was created and a future date. Simulations are run to validate whether such changes are admissible to a network.

  An application could use the output from the Calendaring Module to run as input to the simulations run by the Optimization and Prediction Module.
- Optimization and Prediction Module—Enables APIs to execute global and tactical optimizations, model traffic-engineered LSPs, and perform failure analysis. These simulation tools enable you to model and predict how the network will react to a specified list of changes, as well as suggest additional changes that would produce a more optimal network.

  Once satisfied with the optimizations and changes, you can choose to send this new model to the Network Modeler Module.
- Deployer Module—Compares the current network model and the new, updated network model to create a change plan of the differences, and then deploys LSP changes to the network via an OSC or NSO controller. If configured, the applications can be sent deployment status so they can determine when deployments are completed.

Note that the Collector Module uses a Collector server and a WAE Network Interface (NI) server, while the remaining modules use the WAE Core server.

Figure 1-1 WAE Platform Workflow



Configuration

This guide describes the following configurations.

- Collecting Network Information—Describes the differences between the collection methods and helps you identify which one is best for you.
- Deploying Network Changes—Describes how to configure WAE Core server resources, as well as how to configure the REST and Thrift APIs by modifying configuration files. It also describes how to configure the deployment of LSPs.

Environment Variables

This guide uses the following environment variables:

- `$CARIDEN_ROOT` and `$WAE_ROOT`—Location of the installation. By default, this is `/opt/cariden`. These terms are interchangeable.
- `$CARIDEN_HOME`—Directory in which the WAE Design, WAE Live, and WAE Collector executables and binaries are installed. The default is `/opt/cariden/software/mate/current`.
- `$WAE_HOME`—Directory in which the packages are installed. The default is `/opt/cariden/software`.

Collecting Network Information

Collection Overview

The WAE collection process runs through a list of sequential tasks that discover network topology (including IGP, node, interface, LSP [SNMP], and PCEP information) and gather traffic statistics (or poll for traffic).

There are various methods in which to collect data (see Collection Methods). Regardless of the method used, the basic unit of data storage that is produced after the collection is called a plan file. Plan files contain network information at a specific time and are used by all WAE applications.

Since typical collection is done through a series of sequential tasks, you might want to have a few tasks performed asynchronously. You can configure WAE to enable continuous traffic polling and PCEP collection after the initial collection by using the WAE NI server. Configuring this feature is discussed in more detail in the following sections:

- Configure Continuous Polling and Collection in the WAE Collector UI
- Configure Continuous Polling and Collection Using Manual Collection

For more information on collection components (WAE Collector, WAE Network Interface, and the data topology workflow, see Data Flow After Collection.

Snapshot Files

Note

- Do not edit snapshot files if you are collecting network information using the WAE Collector UI.
- For snapshot file examples, see Appendix A, "Snapshot Examples."

The collection process uses configurations defined in snasphot files. The snapshot files consist of a .txt and an optional .inc file, and are located in the `$CARIDEN_HOME/etc` directory. Together, these files enable you to customize tasks that define how your network is discovered and modeled.

The `snapshot.txt` file contains *tasks* that are defined in the `snapshot.inc` file through a series of CLI tools. These tasks and their .inc definitions determine what network information is collected and how the network is modeled. The `snapshot.txt` file also defines environment variables called by the CLI tools in the in the `snapshot.inc` file, thus removing the need to manually update these variables more than one time if you reconfigure your network.

- Tasks in the `snapshot.txt` file are defined in the `snapshot.inc` file. These tasks are performed in the order in which they are sequentially listed in the `snapshot.txt` file.
- Variables `$(variable_name)` in the `snapshot.inc` files are defined in the <ENVIRONMENT> table of the `snapshot.txt` file.
- If there are multiple `snapshot.inc` files, they are executed in the order in which they are listed in the <ENVIRONMENT> table.
- If there are nested `snapshot.inc` files, they are executed in the order in which they are listed in the parent `snapshot.inc` file.

Typically, you need only to customize the `snapshot.txt` file, which contains all the steps needed to perform a typical network discovery. The default `snapshot.inc` file contains details of how each CLI tool is called to execute each task, and can often be left as is.

snapshot.txt

Note There are different types of snapshot.txt files; for example, snapshot_augment_collector.txt and snapshot_hardware_inventory.txt file. They all provide the same type of information and are generally referred to as snapshot.txt files throughout this document.

The `snapshot` tool reads the `snapshot.txt` configuration file to determine the following:

- The discovery environment, such as where to store the data, log files, and debug information (see Environment Variables).
- Which discovery tasks to perform.

Environment Variables

The <ENVIRONMENT> table defines numerous variables that are frequently called by tasks defined in the `snapshot.inc` file. By defining them here, you can avoid the repetition of entering them multiple times. The snapshot.txt file itself contains a description of each of these variables.

Snapshot environment variables apply to the snapshot process only, and are unrelated to host environment variables.

Example: Almost all the tasks call a `work_dir` variable to define the location in which to store the snapshot data.

- In the `snapshot.txt` <ENVIRONMENT> table, you could define the following:

```
home_dir /opt/cariden
work_dir /$(home_dir)/work
```

- In the `snapshot.inc` file, define that all tasks put their output in `$(work_dir)`.

Each parameter must be separated from its value by a TAB. At minimum, you must define the following variables in this table:

- `unique`
- `home_dir`
- `collector_url` (if getting a plan file from the Collector server or WAE Network Interface (NI) server)
- `seed_router` (manual collection only)
- `igp`

- Ensure `isis_level` or `ospf_area` is properly configured, depending on the `igp` setting

You can define your own environment variables for snapshot tasks that you create. However, if using the augmented method, you cannot create environment variables that use the same name as those that are applicable only to the manual collection method. To avoid this error, you could compare snapshot_augment_collector.txt to the `snapshot.txt` file to determine names you must avoid using.

snapshot.txt Tasks

The `snapshot` tool reads the `snapshot.txt` configuration file to determine which WAE Collector tasks to perform. The tasks are organized into four high-level tables, each of which contains a list of available tasks for the discovery process to perform.

snapshot.txt Task Type	Description
<DISCOVERY_TASKS>	Define what type of information to collect, such as IGP database, nodes, MPLS LSP paths, and more.
<POLLING_TASKS>	Define which traffic statistics polling functions to perform.
<FLOW_TASKS>	Define whether to collect NetFlow data and related flow measurements.
<ANALYSIS_TASKS>	• Simplify and arrange nodes and sites in the network plot. • Create and initialize a mesh of traffic demands.
<ARCHIVE_INSERT_TASKS>	Insert the completed plan into an existing archive repository.

Each default task is either enabled (no comment symbol [#]) or disabled (with a comment symbol). To enable a task, remove the comment. Conversely, to disable a task, add a comment to the beginning of its line.

Each task is customized and defined in the `snapshot.inc` file through a series of CLI tools. For information, see snapshot.inc. The `snapshot` tool executes the tasks in the order in which they are listed in the `snapshot.txt` file.

You can remove tasks, and you can add any task (with any name) provided you also reference and define it in the `snapshot.inc` file.

snapshot.inc

You can further customize the snapshot discovery process by adding one or more uniquely named `snapshot.inc` files to the <ENVIRONMENT> table in the `snapshot.txt` file. These `snapshot.inc` files define the behavior of each task that is called by the `snapshot.txt` file. Figure 2-1 shows an example.

- The order of the tasks defined in the `snapshot.txt` file is the order in which they are executed. The order of the task definitions in the `snapshot.inc` file does not matter.
- The `snapshot.inc` files are executed in the order in which they are listed in the <ENVIRONMENT> table.
- If there are nested `snapshot.inc` files, they are executed in the order in which they are listed in the parent `snapshot.inc` file.

The parameters used to call these tasks are listed in an individual task table (Table 2-1). The parameters used for the CLI tools within the tasks are listed in an associated options table (`<options-name>` in Table 2-2).

Within each table, references are made to variables defined in the `snapshot.txt` <ENVIRONMENT> table using the format `$(variable_name)`.

Some tasks can copy intermediate files to a debug folder by calling a `postcmd` after the main tool is called.

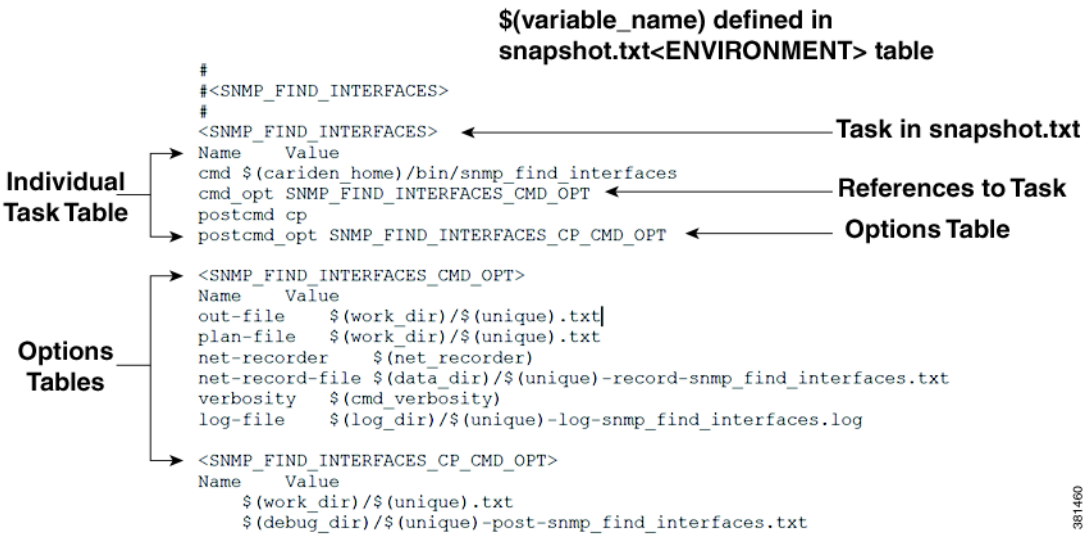
Table 2-1 Individual Task Table

Name	Value
cmd	Fully-qualified name and path of the CLI command to execute. You do not need to change this command during customization.
cmd_opt	Name of the table that defines the command options. You do not need to change the name of the table to change the command options. Instead, edit the contents of the options table by this name.
cmd_success	Determines what exit codes constitute a success for the command. The snapshot process terminates if the command is unsuccessful. 0 = successful, and 1 = unsuccessful
precmd	Fully-qualified name and path of a command to execute before the CLI command. For instance, it can be a task to prepare for the CLI command.
precmd_opt	Name of the table that defines the pre-command options. You can use any name, but its name must match the name of the table that defines the options.
postcmd	Fully-qualified name and path of a command to execute after the CLI command. The default is a Linux `cp` command that copies intermediate files to the debug directory (debug_dir).
postcmd_opt	Name of the table that defines the post-command options. You can use any name, but its name must match the name of the table that defines the options.

Table 2-2 Task Options Table

Name	Value
<option-name>	Value of the option. These are name-value pairs, and you can have as many entries as needed for the command. You can use environment variables to construct filenames. Example: $(work_dir)/$(unique).txt

Figure 2-1 Example Task Defined in snapshot.inc



Launch and Validate snapshot

The `snapshot` tool is located in the `$CARIDEN_HOME/bin` directory. To launch the snapshot tool, enter the following command:

```
snapshot -config-file $WAE_HOME/etc/snapshot.txt
```

You can launch `snapshot` manually or schedule it for periodic operation with a `cron` job. The usual process is to create a `$CARIDEN_ROOT/archives` directory and have the newly discovered plan files saved to it. If you run `snapshot` manually, the resulting plan is placed in the `$CARIDEN_ROOT/work` directory.

If you make changes to either of the snapshot files, we recommend that you initially run the snapshot with the `-dry-run` and `-verify-config` options.

A message of Success after running the tool means the snapshot process successfully executed the tasks identified in `snapshot.txt`. If this is your first time running snapshot, we recommend that you review files in the `$CARIDEN_ROOT/logs` directory for errors and warnings. If you find them, check the `$CARIDEN_ROOT/logs/debugs` directory to see if you can resolve them. You likely need to tweak the authentication, network access, or snapshot configuration file. Common errors include the following:

- Routers inaccessible due to authentication errors, such as incorrect communities.
- Routers not responding or returning incomplete data due to timeouts or other access errors.

When scheduling the `snapshot` tool to run repeatedly and storing plan files into an archive, it is useful to check periodically that the plan files are still valid. Following are a few ways to verify a plan file:

- Look for errors and warnings in the to the `$CARIDEN_ROOT/logs` directory; for example, using `grep`.
- Check the `$CARIDEN_ROOT/work` directory to verify the plan file was created.
- Open the plan file in the WAE Design GUI.

Interval Collections and Continuous Polling

Note For information on how to configure continuous collection and traffic polling, see:

- Configure Continuous Polling and Collection in the WAE Collector UI
- Configure Continuous Polling and Collection Using Manual Collection

Through the WAE Collector UI, you have the option to run collection based on intervals or to combine that with continuous polling.

- Interval collections—This method polls traffic twice during the collection window. The traffic statistics for those two time periods are averaged and added to the plan file as measured traffic. The amount of time for each polling interval is set using the Counter Polling Period field on the Continuous Poller page.
- Use continuous polling—This method polls the traffic continuously. The amount of time for each polling interval is set using the Counter Polling Period field on the What To Collect page. The time window over which the traffic rate is averaged is set in the Default Time Window field. The amount of traffic added is the average traffic for the specified time window at the moment when the plan file is generated.

  Example: Counter Polling Period is 60 seconds. Default Time Window is 15 minutes. Every 60 seconds traffic is polled and added to the plan file. The amount of traffic added is the average traffic for the last 15 minutes at the moment when the plan file is generated.

Collection Methods

There are three main collection methods:

- Basic Collection—Configure basic network collection using the WAE Collector UI. There is no need to edit the snapshot files. The snapshot files are automatically configured based on the entries you define in the WAE UI.

  Note If you edit the snapshot files manually and later decide to use the WAE Collector UI, any configurations made with the WAE Collector UI will overwrite the snapshot files.

- Augmented Collection—Configure basic network collection using the WAE Collector UI and then add more tasks and options using augmented snapshot files (snapshot_augment_collector.txt and snapshot_augment_collector.inc). This method retrieves a plan file for use in WAE Design and WAE Design Archive. Optionally, it enhances the plan file with additional collection, and enhances the plan file with modeling information, such as demands. Examples include parsing configurations for explicit LSP paths, collecting multicast traffic, and collecting flow traffic.

  Note If you only want to build a network model that includes the creation of demands after collection, copy a template plan file into a newly generated plan, and store the resulting plan file into an external plan file archive, then use the network Model Manager. For more information, see Chapter 3, “Viewing a Network Model.”

- Manual Collection—Configure collections using only the CLI. This method is used for advanced configurations that are not supported by the other collection methods. Examples include collection directly from configuration files, multi-networking collection, and collection from Alcatel-Lucent's 5620 Service Aware Manager (SAM) server.

See the following table to determine which collection method to use. Consider what it is you are trying to discover and what the application needs are.

Note The Augmented Collection column indicates features that are not supported if you run only Basic Collection using the WAE Collector UI. To enable these features, you must run Augmented Collection after Basic Collection is completed (see Collecting Information Using Augmented Collection).

Table 2-3 Collection by Configuration Method

	Basic Collection Using WAE UI[1]	Augmented Collection	Manual Collection
Uses SNMPv2c authentication	x		x
Uses SNMPv3 authentication	x		x
Directly discovers nodes using system IPv4 addresses	x		
Collects OSPF and IS-IS IPv4 topologies	x		x
Collects OSPF and IS-IS IPv6 topologies			x
Collects BGP LS topologies			x
Collects node properties	x		x
Collects interface properties, including TE extensions	x		x
Collects interface queues	x		x
Collects Segment Routing LSPs			x
Collects interface traffic based on egress shaping rate			x
Collects SRLGs		x	x
Discovers BGP peering	x		x
Continuously polls traffic statistics (requires the WAE NI server)	x		x
Continuously collects LSPs (requires the WAE NI server)	x		x
Collects basic RSVP LSP properties	x		x
Collects RSVP LSPs with multiple paths or named paths (EROs)		x	x
Collects LAG[2] ports	x		x
Collects RSVP LSP affinities			x
Collects Multicast		x	x
Collects VPNs	x (Layer 3 only)	x	x
Collects LDPs		x	x
Collects flow traffic		x	x
Collects topology from config files			x

Table 2-3 Collection by Configuration Method (continued)

	Basic Collection Using WAE UI[1]	Augmented Collection	Manual Collection
Collects Layer 1 and Layer 3 information			x
Can build network models after the collection process, including the creation of demands		x	x
Collects hardware inventory		x	x
Collects multiple networks			x
Collects from SAM server			x

1. This table does not include the advanced configuration options available in the Collector UI. Additionally, all collections are dependent on licenses and what you have configured for collection.
2. Vendors have different names for LAGs. For instance, Cisco IOS uses the term *EtherChannel* (port-channel interface), Cisco IOS XR uses the term *link bundling* (bundle-ether interface), and both Juniper and Alcatel-Lucent use the term *LAG*.

Plan Files

All WAE applications use plan files produced by WAE Collector. Plan files capture all relevant information about a network at a given time, and can include topology, traffic, routing, and related information. How and where plan files are created depends on the collection method and what is configured in the snapshot files.

- From the WAE Live UI, you specify where the application is to get its plan files: either from a server or from an external plan file archive that is used by the augmented and manual discovery methods.
- The WAE Design Archive UI uses the plan files that are stored in the external plan file archive.
- The WAE Design GUI can access plan files from either the plan file archive that is internal to WAE Live or from the external plan file archive simply by telling the GUI which remote server to access. The primary use for this application to access the plan file archives is to (1) create and update templates for use in WAE Live and WAE Design Archive, or (2) simulate traffic based on discovered data when designing and planning networks using WAE Design.

WAE Collector and Archives

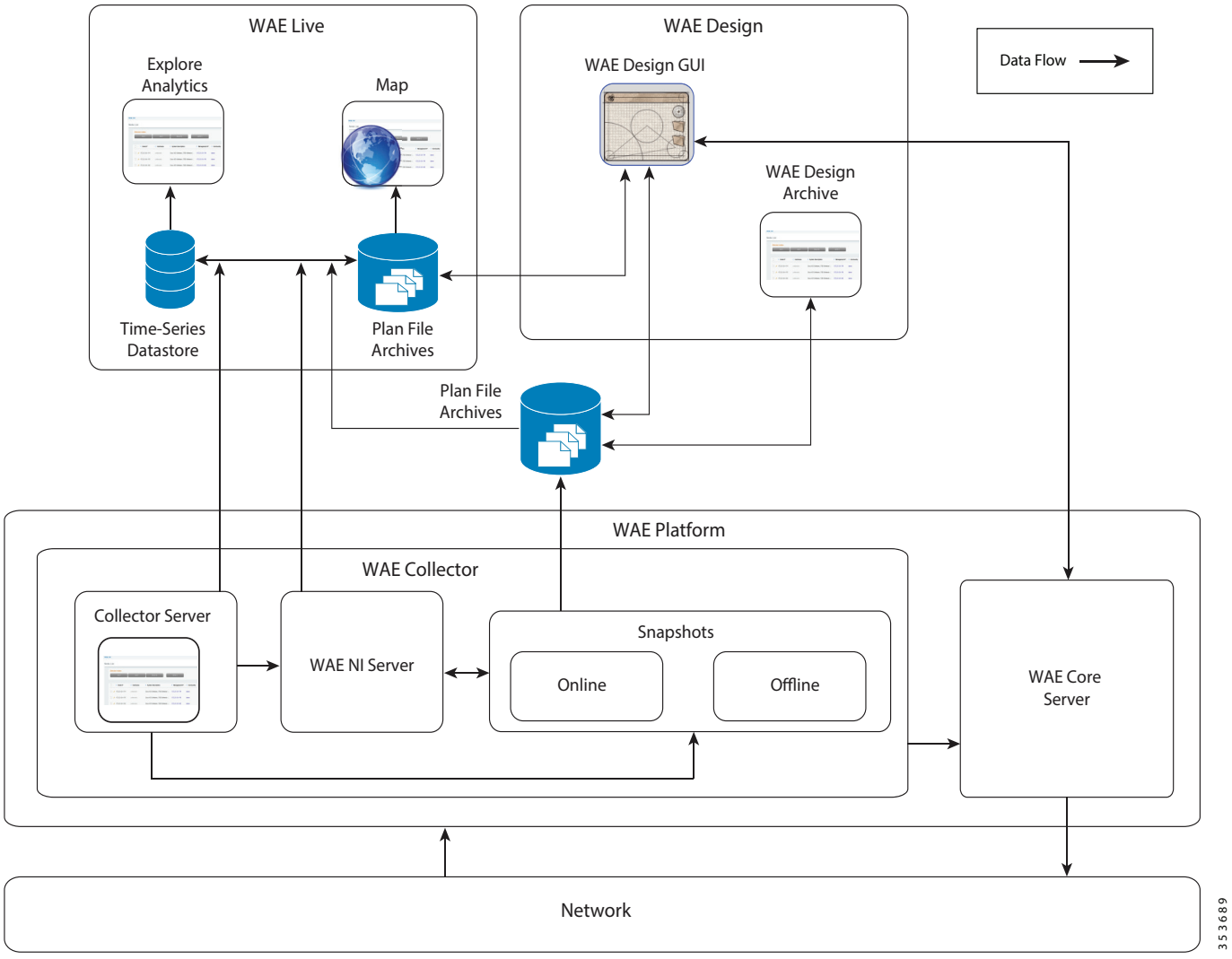
The first step is for WAE Collector to discover the network and create a plan file that represents your network.

- The Collector server, which is configured only through the WAE Collector UI, discovers the network at user-defined intervals to create and store the plan files on that server. The plan files reside on one of these servers until either WAE Live or a snapshot process requests them.
- If using the augmented method of discovery, the snapshot uses a plan file generated by the Collector server, and then adds other aspects of the network (such as Multicast). A common use case for augmented snapshots is to add modeling elements, such as demand meshes, and to perform demand deduction for use in applications. The resulting plan file is sent to an external plan file archive.
- If manually discovering the network, either through online or offline means, snapshots run at user-defined intervals and distribute the plan files to an external plan file archive repository. Optionally, you can configure the continuous polling of traffic statistics.
- WAE Collector sends updated plan files to the Network Modeler Module.

Note Optionally, you can configure the Collector server or a snapshot to push the plan file, as well as the access and authentication files, to the WAE NI server for the continuous polling of traffic statistics or for the continuous collection of LSPs. From a snapshot, you can also pull a plan file from the WAE NI server.

Table 2-3 shows how data flows between WAE Collector and the archives, and how data flows between the archives and the WAE applications. This diagram does not depict template flow, where *template* is a plan file containing the visual aspects that display the network in the application interfaces. For information on templates, see the *Cisco WAE Network Visualization Guide*.

Figure 2-2 Data Flow After Collection



Using Collections

You can use collections in the following ways:

- If using WAE Live, configure it to collect from the appropriate source and specify the Map archive location. For information, see the *Cisco WAE Live Administration Guide*.
- Use the WAE Design GUI to update a template for use by the applications. For information, refer to the *Cisco WAE Network Visualization Guide*.
- To verify the plan file collection has been set up correctly, open the plan file from the application you are using.

Collecting Basic Information Using the WAE Collector UI

Note Snapshot files are automatically configured using the WAE Collector UI. If you have previously performed a manual collection or edited the snapshot files, the configuration will not be applied. You will need to redo the configuration through the WAE Collector UI.

The WAE Collector UI enables you to configure the collection of basic network data for different networks. The Collector server handles router access and authentication, while enabling you to configure and schedule collection, and troubleshoot any issues. In many cases, the plan file produced can be used directly by WAE Live.

The WAE Collector UI predominantly uses the Collector server. You can also start the WAE NI server and thereafter connect to it through the WAE Collector UI. Then you can delegate traffic polling for objects and the continuous collection of LSPs to the WAE NI server.

Once the collection finishes, WAE Collector creates a plan file.

- If continuous polling is not running and if LSPs are not being continuously collected, the plan file is generated based on the completion of a collection as configured from the Schedule page. To retrieve the plan file, access it from the Collector server.
- If continuous polling is running or if LSPs are being continuously collected, the plan file is generated on demand, such as when the WAE Live application requests it. Additionally, the WAE NI server caches the plan file at regular intervals. To retrieve the plan file, access it from the WAE NI server.

New nodes are added when they are discovered. Nodes that are removed from the network (manually or through failure) are set to inactive. This inactive state is kept for a user-configurable time, after which the nodes are removed from the collection.

Workflow for Collecting Basic Information Using the WAE Collector UI

The initial workflow consists of the following steps. You can return to any of these steps at any time to change the configurations.

If you have not yet configured a node list used for collection or if you restarted WAE Collector, a setup wizard is available to lead you to the required Setup pages.

Note You are notified if another user is accessing the WAE Collector UI when you log in. Note that any changes you make will affect the other user's configuration, and vice versa.

Table 2-4 WAE Collector UI Collection Configuration Workflow

Step	Task	Description
1	Configure Node Discovery	Define how nodes are discovered and configure global default device SNMP community and login credentials.
2	Configure Node Access	Define the management IPs.
3	Configure Additional Node Access Profiles	(Optional) Configure and apply additional node access profiles. For example, apply a node access profile for nodes depending on vendor or model.
4	Configure Node Inclusion	(Optional) Set global rules for including and excluding nodes from being collected.
5	View and Manage the Node List	View and create override rules for specific nodes to which the global default credentials do not apply. You can also exclude nodes and apply profiles here.
6	Configure What to Collect	Configure which objects and traffic to collect, and set the counter polling period.
7	Configure Continuous Polling and Collection in the WAE Collector UI	Configure continuous traffic polling and collection using the WAE Network Interface server.
8	Schedule the Collection	Schedule how frequently you want to collect network data.
Other Tasks		
	Add Additional Networks for Collection	Add additional networks for collection.
	View Collection and WAE Collector Server Status Details	View local Collector server information.
	View Collector Server Logs	View collection events and messages.
	Save or Load Configurations	Save or load configurations that were made using the WAE Collector UI.

Configure Node Discovery

The initial step to configuring collection is to define how nodes are discovered by reading the IGP database of a seed router or by specifying a list of system IP addresses. You can combine these methods to populate the node list.

Note These rules can be overwritten on a per-node basis using the Node List.

For Node Discovery field descriptions, see Table 2-5.

Step 1 From the WAE UI, select **WAE Collector.**

Step 2 If you have multiple networks, select the applicable network that you want to configure collection for. The network icon is located next to the WAE Collector menu item.

Step 3 Choose **Setup > Node Discovery > Default Credentials** tab.

Step 4 Configure global default seed router credentials and click **Apply**.

Step 5 Select the IGP Discovery tab and configure IGP discovery settings and click **Apply**.

WAE Collector communicates with a seed router using its management IP address. The node list is populated with all nodes in the IGP database of the seed router. All WAE Collector interactions applied in the UI work from this node list.

Step 6 Select the Direct Node Discovery tab and configure direct node discovery using system IPv4 addresses and click **Apply**.

WAE Collector uses a list of user-specified system IPv4 IP addresses to discover nodes that may or may not be in the IGP database. SNMP is used to find and poll nodes and interfaces. Other objects, such as LSPs and VPNs, cannot be found using this method. One use case is for discovering L2 switches that reside within a router's domain, but are not listed in the IGP database.

Node Discovery Field Descriptions

Table 2-5 Node Discovery Field Descriptions

Node Discovery Tab	Description
Default Credentials	
SNMPv2c Default RO community	This field entry is required. Enter the community string that acts as a password. It is used to authenticate messages sent between the node and the seed router. You can specify SNMPv3 credentials for specific nodes by creating additional profiles and applying the profile to specific nodes on the node list. For more information, see Configure Additional Node Access Profiles and View and Manage the Node List.
Login	If you want to use a specific username and password to log into devices, select **Specify** and enter the appropriate credentials. If not, select **Disable**.
Security	This field entry is required. Enter a master password to enable you to de-encrypt the authentication file. The password must contain a lowercase and an uppercase character, a special character, and a number.
IGP Discovery	
Discover using IGP Database	Check the check box to use an IGP seed router to discover nodes.
Seed Router Management IP	Management IP address of the seed router used for all collections. The node list is populated with all nodes in the IGP database of the seed router.
Select IGP	**OSPFv2**—Select if collecting an OSPF database. **IS-IS**—Select if collecting an Intermediate System-to-Intermediate System (IS-IS) database. **Note** Node names are available if using IS-IS. You must log into the seed router to discover IS-IS.
IS-IS Level	This option is only available if IS-IS was selected as the database. Select whether to use Level 1, Level 2, or both. If a single Level is selected, the seed router must belong to that level. If selecting both, WAE Collector attempts to log into other routers as necessary, using the same credentials as the seed router, to assemble the nodes from both levels.

Table 2-5 Node Discovery Field Descriptions (continued)

Node Discovery Tab	Description
Select OSPF Area	**Specify**—Select if collecting from a single OSPF area and enter an area ID. The seed router must belong to the area specified. **All**—Select if collecting from all OSPF areas. In this case, WAE Collector attempts to log into all Area Border Routers (ABRs) using the same credentials as the seed router to assemble the nodes from each area. **Note** Unlike the IS-IS database, the OSPF database does not contain node names. Node names will only be available in the node list after SNMP access to each node is established using the Node Access page.
Initial Authentication	Select whether to log into the seed router or use SNMPv2 to access it. If discovering IS-IS, you must select Login.
Login Session Type	Select which login protocol to use: SSH or Telnet. The SSH protocol is more secure and is recommended, if available. The Telnet protocol does not encrypt the username and password.
Use Backup Seed Router	Check the check box to identify whether to use a backup router if the seed router becomes unreachable.
Backup Management IP	Management IP address of the backup router should the seed router not be reachable. This is required if "Use backup seed router" is enabled.
Direct Node Discovery	
Discovery using System IPv4 Addresses	Check the check box to use IP addresses to discover nodes. Enter one or more IPv4 addresses separated by commas. You cannot specify a subnet range.

Configure Node Access

The Node Access page enables you to define the management IPs, SNMP communities, and if necessary, login credentials used by WAE Collector to reach the nodes. The options on this page enable you to reach nodes that could not be reached using strictly the seed router defined on the Node Discovery page. Regardless of whether you are using login or SNMP to reach the seed router, you can use another mechanism to reach the other routers. For instance, you can configure SNMP to reach the seed router and use login to reach the other routers.

The nodes' management IP can be set to one of two rules: set the management IP address to be the same as the node ID (router ID) or replace the node IP address prefix with a user-defined IP prefix.

If discovering multi-hop BGP or if adding login tasks through the Advanced Configurations tab on the What to Collect page, you must enable login through the Login Access option. WAE Collector collects basic BGP information from SNMP, but may need to log into specific routers if multi-hop BGP is configured. You can optionally set these to be the same credentials as used by the seed router.

When the configuration is applied, whether a node is reachable is indicated in the SNMP and Login columns of the Node List table.

You can also configure additional credential profiles using the Additional Access Profile page.

Step 1 From the WAE UI, choose **WAE Collector > Setup > Node Access**.

Step 2 Select and enter the appropriate information. See Table 2-6 for field descriptions.

Step 3 Click **Apply**.

Step 4 (Optional) Configure additional node access profiles. For more information, see Configure Additional Node Access Profiles.

Note

- You can edit this page at any time. Doing so changes how nodes are reachable.
- The global node access rules can be overwritten on a per-node basis. If this is the first time you are setting up the node list, continue to Configure Node Inclusion.

Node Access Field Descriptions

Table 2-6 Fields in Node Access

Field	Description
Management IP	
Same as node IP	Select if the node management IP address is the same as the node IP address.
Replace node IP address prefix with	Select this option if the management IP address can be derived by changing the IP prefix. Enter the node IP address in the first field, and enter the substitution pattern in the second. **Example:** The node IP addresses are in the range 5.6.7.8/24, and the management IP addresses are in the range 5.6.77.8/24. Thus, 5.6.7.8.1 maps to 5.6.77.8.1. For example, to apply this rule, enter: 5.6.7.8/24 > 5.6.77.8/24

Configure Additional Node Access Profiles

You can configure and apply additional node access profiles to several nodes using the Additional Access Profile page. The creation of node access profiles, for example, allows you to easily apply specific credentials to a group of nodes that belong to a certain device model or vendor.

Step 1 From the WAE UI, choose **WAE Collector > Setup > Additional Access Profile**.

Step 2 Enter appropriate credential information (see Table 2-7).

Step 3 Click **Save**.

Step 4 To apply the new profile to specific nodes:

a. From the WAE UI, choose **WAE Collector > Node List**.

b. Check all nodes that you want to apply the new profile to.

Note To sort nodes, click the appropriate column heading.

c. Click **Edit**.

d. From the drop-down fields, select the new profile to apply to the nodes.

e. Click **OK**.

f. To verify that the profile has been applied correctly, click **Test**. View the node list to see if the credential status icons changed for the selected nodes.

Additional Access Profile Field Descriptions

Table 2-7 Fields in Additional Access Profile

Field	Description
Choose Profile	If one exists, choose a profile to edit.
Profile Name	Enter a name for the profile.
SNMP	
SNMP access options	Select either **SNMPv2c** or **SNMPv3**.
SNMPv2c Default RO community	Enter the community string that acts as a password. It is used to authenticate messages sent between the node and the seed router.
SNMPv3 Default Credentials	
Security Level	Select one of the following: • **noAuthNoPriv**—No authentication or privacy protocols are used. • **authNoPriv**—Authentication protocol is used. • **authPriv**—Both authentication and privacy protocols are used.
Username	Enter the username that is configured for the SNMP agent.
Authentication Protocol	Select **MD5** or **SHA** protocol used for authentication.
Authentication Password	Enter the password used for authentication. The password must be at least 8 characters long.
Encryption Protocol	Select **DES** or **AES-128** encryption.
Encryption Password	Enter the password used for encryption.The password must be at least 8 characters long.
Login	
Login access options	If you want to use a specific username and password to log into devices, select **Specify** and enter the appropriate credentials. If not, select **Disable**.

Configure Node Inclusion

The Node Inclusion page enables you to set global rules for including and excluding nodes from being collected. You can edit this page at any time. Doing so changes global rules for whether nodes are included or excluded from being collected. The exclusion rule always takes precedence.

All rules are set using regular expressions. Use the inclusion or exclusion options that make it easiest for you to define the necessary hostnames. For instance, inclusion rules can be useful when you are discovering more nodes than you have available licenses, or when you are only interested in collecting a subset of the nodes.

Example: These are the nodes.

- core1-atl2.acme.com
- core2-atl2.east7.com
- dist1-atl2.acme.com
- core2-atl1.acme.com
- core1-chg1.acme.com

Section	RegEx	Result
Include only nodes	.*	Include all five nodes
Exclude any nodes	^dist.*|.+east.*	Exclude all nodes with a prefix of "dist" or that contain the string "east." The excluded nodes are core2-atl2.east7.com and dist1-atl2.acme.com.

Step 1 From the WAE UI, choose **WAE Collector > Setup > Node Inclusion**.

Step 2 Enter regular expressions to filter what to nodes to include in the given area. To exclude nodes, click the **In addition, exclude any nodes with name matching regular expression** box and enter the regular expression in the given area.

Step 3 Click **Apply**. The updates are displayed in the Include column in the Nodes List table. For more information, see View and Manage the Node List.

The global node inclusion and exclusion rules can be overwritten on a per-node basis. Thereafter, if you continue to see a need to create per-node overrides, use the Node List page.

View and Manage the Node List

The Node List page allows you to create override rules for specific nodes to which the global default credentials do not apply.

The Node List table displays all nodes available to be used in the collection process. Use this table to determine whether nodes are included or excluded, whether nodes are accessible through SNMP or login, and the properties of each node.

The Node List page also provides a means of creating per-node rules that override the global ones. After configuring your global rules, use this editing feature to fine-tune the list of nodes collected.

Each row shows the node attributes, access status, and collection status. This is where you manually override the management IP, SNMP community, or login settings for nodes when the global rules do not succeed. You have the option of specifying explicit values, or you can scan a subnet trying different SNMP communities to find the correct IP address. This scan is useful when you enter an override rule for one or more nodes.

Note

- Nodes are based on two criteria on whether they are included in the collection or not. One is an exclusion based on global rules, and one is an exclusion based on per-node override rules.

- If the number of nodes discovered is more than the number of licenses available, licenses are allocated based on ascending order of system IP addresses, but all of them are included in the collection. Node license violations are listed at the top of Node List page and on the Status page.

Edit Node Credentials (Override Rules)

When new nodes appear, WAE tries the global community string in combination with the global management IP that were specified in the node discovery setup. If SNMP access fails, you can get information for these failures on the Status page. For more information, see View Collection and WAE Collector Server Status Details.

Once the problem is identified, use the Node List page to run a test to see which nodes are being collected, which ones are not, and which nodes were just installed. The nodes that are failing are the ones for which the global rules are likely not working.

For each failed node, if you know the management IP, the SNMP community string, and/or the login access information for that node, you can override the global credentials. If you do not know this information, you can use the scan feature that is available using the Discover option of the Edit field.

Step 1 From the WAE UI, choose **WAE Collector > Node List**.

Step 2 Check all nodes that you want to edit and apply the same credentials to.

Note If you want to only edit the management IP address of one or more nodes, click the management IP address and edit the cell directly from the table.

Step 3 Click **Edit**.

Step 4 Do the following:

a. Exclude from collection—Select to use existing inclusion rule configured from setup, or choose to exclude or include the node.

b. Edit—Select one of the following:

- **Specify**—Select whether to use global rules or override rules for the selected nodes. Then specify changes to management IP, SNMP community, and login access as needed. The SNMP status in the Node List is set to “unknown” until the next collection runs.
- **Discover**—Enter the subnet to search. Then enter multiple SNMP communities to try in succession. WAE Collector scans a range of management IPs combined with the different communities entered to find a node with an ID that matches the discovered node ID. WAE then scans the entire subnet using the entered communities strings in sequence. WAE then tries to find a combination of management IP and community string that allows SNMP access to a router. If such a combination is found, then SNMP access is used to verify whether the found router is in the node list.

  Example: A subnet contains 256 total addresses, and there are 2 SNMP communities to match. This yields a total of 512 attempts to find a node that matches the combination of the subnet and either of the SNMP community strings.

Step 5 Click **OK**.

Step 6 To verify that the credentials have been applied correctly, click **Test**. View the node list to see if the status icons changed for the selected nodes.

Delete, Test, and Apply Updates to the Node List

Step 1 From the WAE UI, choose **WAE Collector > Node List**.

Step 2 Check all nodes that you want to perform an action upon.

Step 3 Click one of the following:

- **Test**—Before applying the configuration, click **Test** to determine if the selected nodes are reachable and included. If a node is not reachable, change its per-node override rules as needed.
- **Delete**—Removes a node from the Node List.

Note WAE Collector never dynamically removes nodes from the Node List, even those that are no longer found during the discovery process. This avoids losing node-specific configurations of nodes that are removed and then later re-appear in the network. To remove a node from the Node List, you must manually delete it from the list.

- **Apply**—Applies the configuration, which updates the Node List.

Node List Table Columns

The Node List table identifies all nodes available for collection, their properties, and status.

If values are user-configured in the UI, they are color-coded based on how they are configured. If the field is blue, the associated node was derived using the global rules. If it is black, the node was derived from the override rules

Table 2-8 Status Columns

Icon	Include	SNMP	Login	Match
Green check	Include in the collection process	Successful SNMP query	Successful login using the management IP address	A match occurs if the node IP is one of the loopbacks configured on the node or if the node name is identical to the node name informed by SNMP.
Red cross	No license or invalid license, but the nodes are still included in the collection	Unsuccessful SNMP query using the management IP address	Unsuccessful login using the management IP address	There is no match. The node IP is not one of the loopbacks configured on the node and the node name is not identical to the node name informed by SNMP.
Blue cross	Excluded from collection by global exclusion rules	NA	NA	NA

Icon	Include	SNMP	Login	Match
Black cross	Excluded from collection by explicit per-node rule	NA	NA	NA
Gray circle	Not determined	SNMP not attempted	Login not attempted	NA

Table 2-9 Property Columns

Property	Description
Management IP	Node management IP address
Community	Encrypted SNMPv2c community string, which is a text string that acts as a password
Username	Name used by WAE Collector to reach the node
Password	Password used in conjunction with the username by WAE Collector to reach the node
Node IP	Router ID
Hostname	Node ID
Vendor	Router vendor
Model	Router model number
OS	Router operating system and version
Last IGP Update	Most recent timestamp of when the node was included in an IGP collection

Configure What to Collect

After you have verified the node list, the next step is to identify what to collect from these nodes. The What to Collect page enables you to optionally collect properties, traffic, BGP connectivity, and VPNs. The information collected for each object populates the plan file tables for use in WAE applications.

• Basic properties and traffic – Nodes – Interfaces – Interfaces queues – RSVP TE LSPs – LSPs	• BGP connectivity • Layer 3 VPNs and traffic

Node names collected from the network often have long suffixes that are the same for all nodes. This page enables you to remove these suffixes, acting on all nodes in the collection process, making for more readable plan files and WAE Live data. The page also provides a feature that enables you to set how long to keep inactive interfaces and circuits from the plan file, thus keeping plan files up to date.

Step 1 From the WAE UI, choose **WAE Collector > Collection > What To Collect**.

Step 2 Click the **Basic** or **BGP/VPN** tab and enter the applicable information. See Table 2-10 for field descriptions.

Note While all fields are optional, you must choose Interfaces, LSPs, BGP, or VPN to collect any data. After configuring these fields, click **Apply**.

Step 3 (Optional) To configure continuous LSP collection or traffic polling, follow the steps described in Configure Continuous Polling and Collection in the WAE Collector UI.

Step 4 (Optional) To add options to existing commands or add new commands, select the **Advanced** tab. This feature is only for advanced users. Modifying the configuration can break the collection process. New commands must be only for collection purposes. The validation process does not guarantee that the modified configuration will work. Consult your support representative for assistance.

Adding an option to a command that has an option with the same name overwrites the existing one. Therefore, always use unique option names.

If using continuous polling, options added to SNMP_POLL are ignored. If adding login commands, you must enable login through the Login Access option on the Node Access page (see View and Manage the Node List).

Step 5 Schedule the collection. For more information, see Schedule the Collection.

What To Collect Basic Field Descriptions

Table 2-10 Field Descriptions for What to Collect

What to Collect Tab	Description
Basic	
Interfaces	Check the check box to collectively identify each interface. For example, an interface's properties could include its interface name, capacity, IGP metric, and TE metric.
Interfaces: Include Queues	The list of interface queues configured on the router, together with per-queue traffic measurements.
Interfaces: Traffic	Incoming and outgoing traffic on an interface in Mbps.
Interfaces: Counter Polling Period	The intervals (in seconds) between successive traffic counter polls.
LSPs	Check the check box to collectively identify each LSP. For example, an LSP's properties could include its destination, setup bandwidth, and the actual path of the LSP.
LSPs: Non-PCEP LSPs	LSPs that are discovered using SNMP. **Note** To set up continuous collection for these LSPs, check this check box. For information, see Configure Continuous Polling and Collection in the WAE Collector UI.
LSPs: PCEP LSPs	LSPs that are managed by WAE or delegated to WAE. LSP delegation is when a node (router) grants WAE the right to update the LSP attributes on one or more LSPs. **Note** To set up continuous collection for these LSPs, check this check box and enable PCEP collection using the steps described in Configure Continuous Polling and Collection in the WAE Collector UI.

Table 2-10 Field Descriptions for What to Collect (continued)

What to Collect Tab	Description
LSPs: Traffic	Outgoing traffic on an LSP in Mbps. • **Counter polling period**— The intervals (in seconds) between successive WAE Collector traffic counter polls. This value must be lower than the LSP counter update period. **Note** Interval collection polls traffic twice during the collection window. The traffic statistics for those two time periods are averaged and added to the plan fils as the traffic. The amount of time for each polling interval is set here. • **Number of nodes with delayed counter update**—The number of nodes that have counters as specified in the Select Nodes field. Certain router vendors and models do not continuously update LSP polling counters. For accurate LSP polling, WAE Collector needs to know which nodes have delayed counters and what the update period is in order to correctly compute the LSP traffic. Use the LSP section to specify the subset of routers with delayed counter updates, and to specify the update delay. The nodes are defined with regular expressions written to find node IPs, node names, vendors, or OS's. If no value is set, the default is 0 and counters are ignored. • **Select Nodes**—Specify which nodes have delayed counter updates. This is specified using a regular expression match on an LSP property. • **Clear nodes**—Clears selection of nodes with delayed counters, and clears all selections made within the LSP section. • **Counter update period**—The amount of time (in seconds) between updates to the SNMP polling counter. Note this value must be higher than the LSP counter polling period.
Remove Node Name Suffixes	Comma separated list of suffixes to remove from node names. This can make the plan file much easier to read in the applications. **Example:** The suffixes acme.net and acme2.net from all nodes in the collection process. acme.net, acme2.net
Days to Expire Inactive Nodes and Circuits	The number of days an inactive node or circuit remains in the plan file before being removed.
BGP/VPN	
BGP	Check the check box to configure discovery of eBGP peers and neighboring external ASes. If discovering multi-hop BGP, you must enable login through the Login Access option on the Node Access page (see Configure Node Access). WAE Collector collects basic BGP information from SNMP, but may need to log into specific routers if multi-hop BGP is configured. You can optionally set these to be the same credentials as used by the seed router. If a default login is not possible, then configure the login access on a per-node basis from the Node List page (see View and Manage the Node List).
BGP: BGP Peer Protocol	Select to discover eBGP peers and neighboring external ASes. Options include searching for BGP peers based on IPv4 addresses, IPv6 addresses, or both.
BGP: Minimum IPv4 Prefix Length	Minimum prefix length to perform an IPv4 subnet match from 0 to 32.
BGP: Minimum IPv6 Prefix Length	Minimum prefix length to perform an IPv4 subnet match from 0 to 128.

Table 2-10 Field Descriptions for What to Collect (continued)

What to Collect Tab	Description
BGP: Multi-hop Discovery by Login	Log into the routers to discover the hops between them. This login must be specified on the Node List page (see View and Manage the Node List).
VPN	Check the check box to discover VPNs and their traffic.
VPN: VPN Type	Select to discover Layer 3 VPN nodes and their traffic. VPN traffic is polled at the same frequency set in the Counter Polling Period field in the Basic page.

Configure Continuous Polling and Collection in the WAE Collector UI

Note Continuous polling is available only on the Default network.

Continuous polling is available for interfaces, queues, VPNs, and LSPs. Queues and VPNs use the same polling period as interfaces.

The WAE Network Interface (NI) server (see Data Flow After Collection) uses SNMP to continuously poll traffic for discovered objects. The statistics gathered are used to calculate frequent, ongoing traffic averages. This can be useful for keeping traffic statistics up to date during the entire collection process, which generally takes a significantly longer time to run than a single polling period.

The WAE NI server also continuously collects LSPs that are can be deployed by WAE or delegated to WAE. (LSP delegation is when a node (router) grants WAE the right to update the LSP attributes on one or more LSPs.)

When configured through the WAE Collector UI, the WAE NI server generates a plan file every five minutes by default.

Continuous Polling and Collection Example:

Counter Polling Period = 120 seconds (this is configured in the Basic tab)

Default Time Window = 8 minutes

Max Expansion of the Window = 25%

Every 120 seconds traffic is polled. The amount of traffic added to the plan file is the average traffic for the last 8 minutes. If there are insufficient counters to calculate the average, the window is extended by 2 minutes (25% of 8 minutes is 2 minutes).

Prerequisites

- Node discovery has been configured.
- Node List is available.
- The WAE NI service must be running. If it is not running, enter the following command:

```
service wae-ni start
```

For Continuous Collection field descriptions, see Table 2-11.

Step 1 From the WAE UI, choose **WAE Collector > Collection > What To Collect > Continuous Collection** tab.

Step 2 To enable continuous traffic collection, check the **Continuous Polling** check box and enter or accept default options.

Step 3 To enable continuous LSP discovery, check the **LSP Collection** check box.

Note
- This option is available only if LSP (Non-PCEP and/or PCEP) collection is enabled from the Basic tab.
- When this option is enabled, continuous traffic collection is automatically enabled.

Step 4 Connect to the WAE NI server by entering Server Access and Server Configuration details or accept default entries.

Step 5 In the WAE Network Interface Server Access area, click **Apply**.

Step 6 Click the **Refresh** icon to verify the WAE NI server is running and reachable. If it is not, verify that you correctly configured its password, started the server , and correctly entered server information.

Note Status of the WAE NI server does not automatically refresh. You must click the Refresh icon to see the latest status.

Step 7 In the main Continuous Collection tab, click **Apply**.

Step 8 Schedule the collection. For more information, see Schedule the Collection.

Continuous Collection Field Descriptions

Table 2-11 Continuous Collection Field Descriptions

Field	Description
Continuous Polling	
Default Time Window	The amount of time, in minutes, over which to calculate (average) the polled traffic statistics. This window (calculation period) starts at the time the plan file is generated and goes backwards to get the statistics. For instance, if the plan file is generated at 8:00 AM and the Default Time Window is 10 minutes, the plan file generated uses statistics from 7:50 AM to 8:00 AM. **Example:** If set to 5 (300 seconds), to determine the incoming packet error rate, WAE Collector takes the average of these incoming packet errors over the last 5 minutes (difference in incoming packet errors over the 5-minute interval / difference in the timestamps of the collections of these readings).
Max Expansion of the Window	There are times in which average statistics cannot be calculated. For instance, router response time might be slow enough that the WAE NI server cannot get sufficient data. This field creates a safety net for such instances by giving the WAE NI server more time from which to collect data. The value is the percentage by which to expand (add to) the amount of time set in the Default Time Window field if no statistics are collected. The lapses in statistics collection do not have to be synchronous for this parameter to apply. **Example:** If the Default Time Window is 10 minutes and the Max Expansion is set to 50%, the window for calculating averages can be expanded up to 5 minutes (50% of 10 minutes) in the event no statistics are available at any time during the 10-minute window.
LSP Collection	
Collection Interval	The amount of time for each polling interval.

Table 2-11 Continuous Collection Field Descriptions (continued)

Field	Description
WAE Network Interface Server Access	
URL	Enter the hostname or IP address of the server that is running continuous polling. If the Collector server and WAE NI server are on the same device, you can use localhost.
Port	Enter the port number of the server that is running continuous polling. The default is 61617.
Username / Password	Enter the username and password that gives you access to the server that is running continuous polling. Both are case sensitive. The default username is "admin," and the default password is "cariden." If the password has changed and you do not know it, contact your administrator or support representative.

Schedule the Collection

Once you have the node list in place and have defined what you want to collect on these nodes, the final step in the configuration process is to schedule the collection and start it.

Note that a collection is also called a *snapshot*. Once a collection instance (snapshot) is stopped, a new collection automatically starts at the next scheduled collection interval unless you are running a single collection. If the Collector server is stopped, the collection process automatically resumes once the server is restarted. If continuously polling the traffic or if continuously collecting PCEP LSPs, that polling or collection is not affected by stopping the Collector server.

The first time you run a collection or if you have made significant changes to the Node List run the collection once and then check the Status page for warnings or errors to determine where you might need to further improve the collection.

Once the collection process is started, the Status and Logs pages are updated with warnings and errors as they occur. The current state is displayed in the top, right of the screen.

Step 1 From the WAE UI, choose **WAE Collector > Collection > Schedule**.

Step 2 Configure the scheduling options or leave the default values, and click **Apply**. See Table 2-12 for field descriptions.

Step 3 To start or end collection, click one of the following buttons:

- **Start**—Starts the collection process using the configured scheduling options.
- **Stop**—Terminates a scheduled collection.
- **Run Once**—Starts the collection, but it only runs one time.

Schedule Field Descriptions

Table 2-12 Field Descriptions for Schedule

Field	Description
Start new snapshot every	Specify how often you want the collection process to run (in minutes). The daily collection times are computed as 00:00 UTC on the hour. For example, if you set this to 16, collection would occur at 16 minutes after the top of the hour, 32, 48, and then again at the top of the next hour.
Collect snapshots	Specify when you want the collection process to run: throughout the day or up to three specified times periods. For example, if you know the network's peak traffic times and you want to run simulations on this traffic in WAE Design, you could collect only at those peak-traffic intervals. To specify a time period select a row, and then move either side of the sliding bar to set the start and end times. Overlapping time periods are not permitted.
Skipped snapshots before terminating	Collection instances might run longer than specified in the Start New Snapshot Every field. To ensure data collection continues, enter a number to specify how many new collection instances (snapshots) to skip before terminating the one that is running. This enables you to prevent multiple collection instan
Collect verbose diagnostics	Check the check box to specify whether to include SNMP recording files. These files are included when using the Downloading Diagnostics feature, which is available on the Status page.
Default log level	Determines the minimum level of severity in the messages that you collect in the log text file. • Fatal—Any error that is forcing a shutdown of the Collector server. • Error—An error that is fatal to the collection process, but not to the Collector server itself, such as the inability to collect an IGP database from the seed router or backup seed router. • Warn—Anything that could potentially cause oddities in the results, such as a switch over from the seed router to the backup router. • Info—Generally useful information such as when the collection process starts and stops. • Debug—Information that is diagnostically helpful. • Trace—Traces the code to find problems.

View Collection and WAE Collector Server Status Details

You can view local Collector server information using the Status page. This page does not report on the status of the WAE Network Interface (NI) server. For all event logs of all servers in an HA environment, go to the WAE Statistics > Events page. For diagnostic and process status information for all servers, go to the WAE Statistics > Diagnostics and WAE Statistics > Processes page, respectively.

Step 1 From the WAE UI, choose **WAE Collector > Collection > Status**.

Step 2 Select one of the following tabs to view collection and status details:

- **Last Snapshot Status**—This tab gives you a quick summary of what was collected in the last collection process (snapshot), as well as the snapshot's duration and whether there were any license violations. If you are running scheduled collections, it displays the next time a collection will run.

Clicking the Download Diagnostic button creates a .zip file containing information to help troubleshoot the last collection by the local Collector server. If calling Cisco for assistance, it is recommended that you e-mail this file to your support representative.

- **Collection Metrics**—This tab shows metrics for all collections for the last 30 days. Daily metrics are kept for the total number of hours data was collected, the number of collections, and whether there were any license violations. Metrics also include the minimum, maximum, and average collection duration, which could be useful for troubleshooting purposes or for adjusting future collection intervals.

  If using the Filter feature to find durations, the increments are h, m, and s for hours, minutes, and seconds, respectively. Do not enter a space between the number and the increment.

  Example: To find snapshots that lasted longer than 15 minutes, select and enter the following.

  Avg Duration Greater than 15m

- **Status Summary**—After each collection process finishes, the Status Summary tab shows the errors and warnings for the most recently completed collection.
  - Node Summary—This table shows errors and warnings that are attributable only to specific nodes, such as an SNMP access failure.

    To read an error or a warning, click the number in the Error count or Warning count cell.
  - Node Independent Issues—This table shows errors and warnings that are not tied to the discovery of nodes, but rather with the collection and post-collection processing steps.

  If you see there are problems, review the Node List to verify nodes are reachable and included. If they are not, try altering either the per-node override rules or the global rules. If you are still not able to troubleshoot and correct the problem, download the diagnostics and send them to your Cisco support representative.

View Collector Server Logs

You can view a list of all errors and warnings since the Collector server was last started. It is a superset of the information that is listed on the Status page, which is relevant only for the last collection.

The information on this page pertains only to the local Collector server. This page does not list logs for continuous polling or for continuous collection of PCEP LSPs. For all event logs of all servers in a local or distributed environment, go to the WAE Statistics > Events page. For diagnostic and process status information for all servers, go to the WAE Statistics > Diagnostics and WAE Statistics > Processes pages, respectively.

To refresh the list of logs without refreshing the browser page, click the **Refresh** button in the top right of the Logs table.

To view logs, go to the WAE UI and choose **WAE Collector > Collection > Logs**.

Save or Load Configurations

A configuration file contains the discovered objects and properties, as well as the configurations used to discover them.

Step 1 From the WAE UI, choose **WAE Collector > Settings > Configuration** tab.

Step 2 Choose one of the following options:

- **Load Configuration**—Overwrites the existing configuration file, and sets the UI settings to those used to configure the saved collection. If needed, you can use this option to load configuration files from the last major release.
- **Save Configuration**—Saves the current configuration file to `<install_directory>/etc/collector/server/configs`. The default installation directory is `/opt/cariden`.
- **Reset Configuration**—Resets all UI settings to their defaults, which includes emptying the node list.

These capabilities can be helpful when performing upgrades or when you need to recover previous configurations.

Configure Collection History

You can configure how many days to keep a collection or how many collections you want to save in the Collector Server. WAE will use the limit that is first reached.

Step 1 From the WAE UI, choose **WAE Collector > Settings > Collection History** tab.

Step 2 Enter the following:

- **Number of last collections**—Saves the specified number of past collections.
- **Max age in days**—The maximum days a collection is saved.

The current disk space storage is also displayed to help you estimate how much data you want to store.

Add Additional Networks for Collection

If you want to configure collection for additional networks, other than the Default network, do the following:

Note Continuous polling can only be performed on the Default network.

Step 1 From the WAE UI, choose **WAE Collector > Default** network icon.

Step 2 From the drop-down list, choose **Network Manager**. The Network Manager page appears.

Step 3 Click **Add**.

Step 4 Enter a network name and click **Save**. The new network should appear under the Networks list. Note:

- The network name cannot contain any spaces or special characters.
- To delete the network or change the network name, click the network under the Networks list.

- The Default network name cannot be changed.

Collecting Information Using Augmented Collection

The augmented collection method extends the plan file that a server creates to include additional collection and modeling for use in WAE Design and WAE Design Archive. If parsing configurations for explicit LSP paths or collecting multicast, LDP, or flow traffic, use the augmented method of collection. To determine the best collection method for your purposes, see Table 2-3 and Collection Methods.

The process begins by configuring and running basic collection (see Collecting Basic Information Using the WAE Collector UI) and then running an augmented snapshot. If you enable continuous collection, then both the Collector server and WAE NI server must continue to run. See Figure 2-3 for a graphical representation of the augmented collection process with continuous collection and polling enabled.

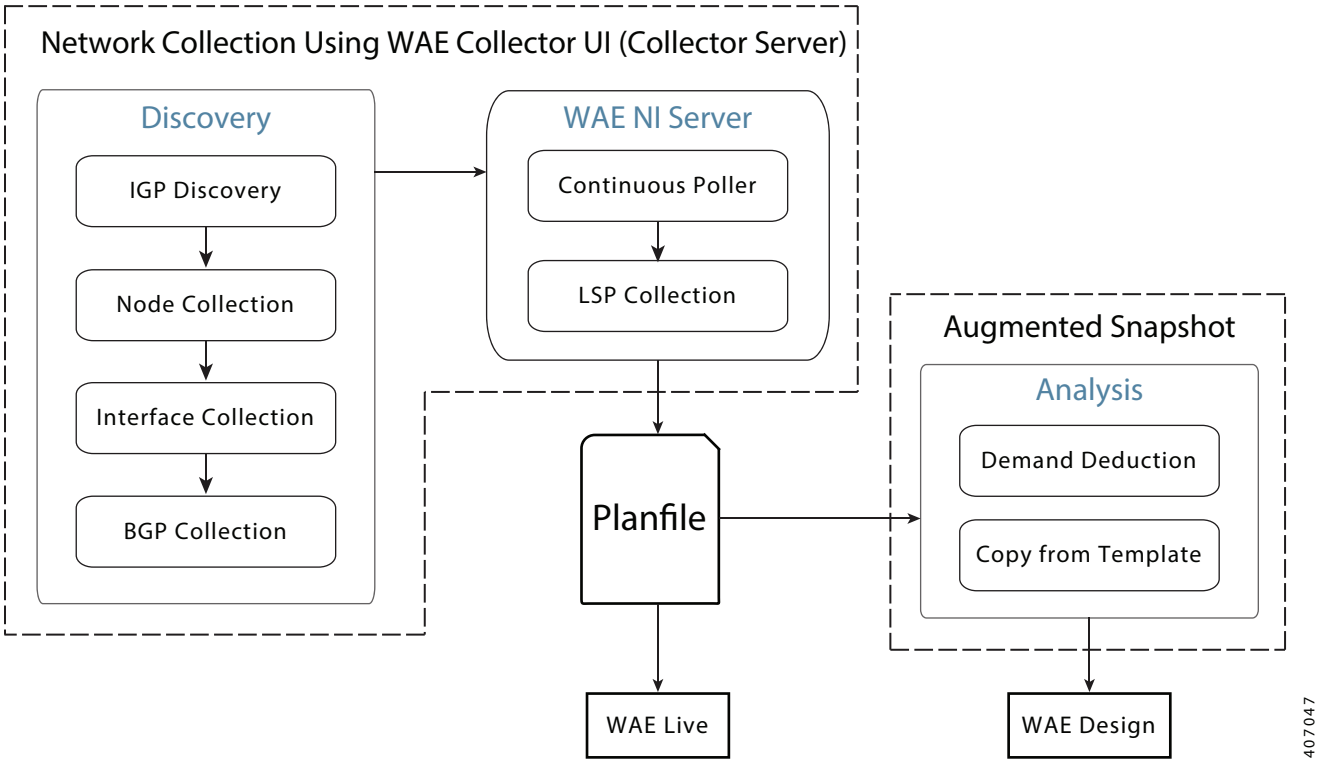
Thereafter, configure the snapshot files to get this plan file from one of these two servers, augment it with additional network data, model the result to visualize the network, and save it in an archive.

One instance of collection must first complete, and thereafter both the server and the augmented snapshot can run simultaneously.

Note If you only want to build a network model that includes the creation of demands after collection, copy a template plan file into a newly generated plan, and store the resulting plan file into an external plan file archive, then use network Model Manager. For more information, see Chapter 3, "Viewing a Network Model."

Note The instructions in this section use the `archive_insert` tool to insert plan files into an external archive. For information on manually inserting plan files into WAE Live, see Appendix A, "Snapshot Examples."

Figure 2-3 Network Collection Process Using Augmented Collection with Continuous Polling and Collection Enabled



Notes and Limitations

The augmented collection method has the following limitations and notable attributes:

- Using augmented snapshots, you cannot collect hardware inventory data, collect data from an Alcatel-Lucent SAM server, or use for multi-network collection. Use the manual method instead.
- This method uses SNMPv2c and SNMPv3 authentication. However, you must use the `mate_auth_init` tool to initiate the authentication file.
- Augmented snapshots can get the plan files from either the Collector server or the WAE NI server. Several of the configuration steps require that you configure one server or the other.
- In the augmented snapshot file, do **not** execute any collection tasks that are available through the WAE Collector UI, including the default ones or any that are configured through the Advanced Config option.
- Do **not** execute SNMP_POLL on interfaces, RSVP-TE LSPs, or VPNs if you are collecting traffic statistics for them through one of the servers.

Environment Variables

- `$CARIDEN_ROOT`—Location of the installation. By default, this is `/opt/cariden`. These terms are interchangeable.

- `$CARIDEN_HOME`—Directory in which the WAE Design, WAE Live, and WAE Collector executables and binaries are installed. The default is `/opt/cariden/software/mate/current`.

Workflow for Collecting Information Using Augmented Snapshots (Augmented Collection)

Before You Begin

We recommend that you back up all your configuration files.

Note If you are only using Augmented Collection to run demand deduction, copy from template, and save to archive tasks, you can use the Model Manager instead.

Table 2-13 Augmented Collection Configuration Workflow

Step	Task	Description / Notes
1	Configure and run network collection using the WAE Collector UI.	See Collecting Basic Information Using the WAE Collector UI. • Access the UI: `https://<Collector_server_IP>:8443/#collector` • If web service is not running, enter the following command: `service wae-web-server start` • Log in to the UI. default username: admin default password: cariden
2	Configure Credentials	Set credentials so that the `collector_getplan` tool can talk to the server from which the snapshot is getting the plan file.
3	Perform Pre-Snapshot Configuration Tasks	Create an authentication file using the `mate_auth_init` tool, optionally edit the network access file, and create two sets of snapshot files for later use.
4	Configure Augmented Snapshot Files	Edit the `snapshot_augment_collector.txt` and `snapshot_augment_collector.inc` files to customize collection.
5	Initialize Archive, Create Template, Run Collections	Post snapshot configuration.

Configure Credentials

You must set credentials so that the `collector_getplan` tool can talk to the server from which the snapshot is getting the plan file. By default, the snapshot authenticates the Collector server.

- You must authenticate the WAE Network Interface (NI) server if using it for continuous LSP collection or traffic polling.
- The credentials file used for the Collector server and WAE NI server must be different.

Before You Begin

Configure and run at least one network collection using the WAE Collector UI (see Collecting Basic Information Using the WAE Collector UI).

Step 1 Run the `collector_getplan` tool once to set the server's credentials for later use in the snapshot files. The only requirement is to use `-set-credentials true`.

```
collector_getplan -set-credentials true
```

The default credential file path, which is configurable, is `$WAE_ROOT/etc/credentials.enc`. To change it, use the `-credentials-file` option.

Example: Set the `-set-credentials` to `true` and change the name of the credentials.enc file.

```
collector_getplan -set-credentials true -credentials-file /opt/cariden/etc/creds.enc
```

Perform Pre-Snapshot Configuration Tasks

Before editing and running the augmented snapshots, you must do the following tasks:

Step 1 Run `mate_auth_init` to create an authentication file (`auth.enc`) used by SNMP and login tools.

```
mate_auth_init
```

This is an interactive tool that first prompts you to choose the SNMP version and the relevant parameters. For information, see Network Authentication.

Step 2 Optional: Customize network access. For information, see Network Access File.

Step 3 For new installations, copy the default snapshot_augment_collector.txt and snapshot_augment_collector.inc files to working configuration files.

```
cp /opt/cariden/software/mate/current/etc/snapshot_augment_collector.txt /opt/cariden/etc
cp /opt/cariden/software/mate/current/etc/snapshot_augment_collector.inc /opt/cariden/etc
```

If this is not a new installation, you can use existing augmented snapshot files in `/opt/cariden/etc` and make modifications noted in this chapter as needed.

Configure Augmented Snapshot Files

Note For information on configuring snapshot .txt and .inc files, see Snapshot Files.

Note A best practice is to add only a few tasks to the snapshot files, run the snapshot, and correct the errors. Then repeat this process until you have built the model of the network that you need.

Step 1 Edit the snapshot_augment_collector.txt file, which contains the collection, polling, modeling, and insertion tasks to perform. This file controls the sequence of execution and also contains environment variables of common values used in the snapshot_augment_collector.inc file.

a. Define the environment variables in the <ENVIRONMENT> section. Each parameter must be separated from its value by a TAB.

   – At minimum, you must define `unique`, `seed_router, home_dir`, and `collector_url`, and preferably the `backup_router`.

- The `collector_url` must be set to the location of the server URL. The default is `https://localhost:8443`, which is to the Collector server. If using the WAE NI server, the port on which it listens for incoming plans is 8086.

  Example: `collector_url https://localhost:8086`

b. If needed, edit the `include` environment variable to read the snapshot_augment_collector.inc file from `$(home_dir)/etc`.

```
include $(home_dir)/etc/snapshot_augment_collector.inc
```

c. Keep COLLECTOR_GETPLAN uncommented as the first task. Either remove or comment out all tasks used in discovering the topology. If you are getting the plan from the WAE NI server that is polling traffic, also remove or comment out all tasks that poll for traffic or collect flows.

Note Do **not** execute any collection tasks that are performed by the Collector server, including the default ones or any that are configured through the Advanced Config option available through the WAE Collector UI. Do **not** execute SNMP_POLL on interfaces, RSVP-TE LSPs, or VPNs if you are collecting traffic statistics for them through one of the servers.

Example:

```
<FLOW_TASKS>
#FLOW_GET
<DISCOVERY_TASKS>
COLLECTOR_GETPLAN
#GET_CONFIGS
#PARSE_CONFIGS
#SNMP_FIND_VPN
<POLLING_TASKS>
#SNMP_POLL
#POLL_LDP
```

d. Define whether to execute flow collection, define which tasks to execute to model the plan file, and define an insert task to specify where to insert the final plan files. Use the comments to enable or disable existing tasks, and add new tasks if needed. At minimum, uncomment the following tasks. For instructions specific to collecting flow data, see Offline Discovery.

- COPY_FROM_TEMPLATE—Copies selected values from the template plan file into the newly generated plan, while preserving network configuration information.
- ARCHIVE_INSERT—Stores the completed plan file in an external plan file archive. This archive can be accessed by all the applications.

Example:

```
<ANALYSIS_TASKS>
#BUILD_MODEL
MATE_CONVERT
COPY_FROM_TEMPLATE
DMD_MESH_CREATOR
DMD_DEDUCT
#MATE_SIM
<ARCHIVE_INSERT_TASKS>
ARCHIVE_INSERT
#ML_INSERT
```

Step 2 As needed, edit the snapshot_augment_collector.inc file to modify and add tools that are to be called from the snapshot_augment_collector.txt file. For information on any tool, refer to its `-help` output. For information on how to edit the snapshot_augment_collector.inc, see Snapshot Files.

For `collector_getplan`, keep `-set-credentials` to `false` so that the snapshot process does not stop to ask for credentials. You only need to set this to `true` once, which you have already done. The `-credentials-file` must match the name that you specified when you first set the credentials (as per Configure Credentials).

Initialize Archive, Create Template, Run Collections

Note Text in <angle brackets> refers to environment variables that you set in the snapshot.txt file.

Step 1 Run `archive_init` to initialize the archive repository into which the plan files will be inserted.

```
archive_init -archive $WAE_ROOT/archives/<unique>-archive
```

For example:
`archive_init -archive $WAE_ROOT/archives/default-archive` (for the default network)

Step 2 If collecting data for WAE Design Archive, use the `archive_config` tool to add the archive repository. At the same time, set up the template directory and template name.

```
archive_config -action add -name <unique> -path $WAE_ROOT/archives/<unique>-archive
-template-dir $WAE_ROOT/data -template-name <unique>-template.pln
```

Step 3 Create an empty template. You can ignore the warnings because the resulting file is an empty template file.

```
echo | mate_convert -plan-file - -out-file $WAE_ROOT/data/<unique>-template.pln
```

Note that WAE Live automatically creates the template.pln from the most recently collected plan file if no template exists. Therefore, for WAE Live, this step is not a requirement.

Step 4 Test the snapshot process by running it as a single tool to collect network data. Check the output for errors, fix them if needed, and rerun this test until it is successful before proceeding.

```
snapshot -config-file $WAE_ROOT/etc/snapshot_augment_collector.txt
```

Step 5 Create a cron job that repeats the process of creating snapshots and inserting them into the archive repository.

Note Both CARIDEN_ROOT and CARIDEN_HOME variables must be defined from within the crontab. You cannot use CARIDEN_HOME=$CARIDEN_ROOT/software/mate/current.

Open the file as follows.

```
crontab -e
```

At the end of the file, add the following lines.

```
CARIDEN_ROOT=/opt/cariden
CARIDEN_HOME=/opt/cariden/software/mate/current
0,15,30,45 * * * * $CARIDEN_HOME/bin/snapshot -config-file
$CARIDEN_ROOT/etc/snapshot_augment_collector.txt 2>&1
```

Collection Network Information Using Manual Collection

The manual collection method uses snapshot.txt and snapshot.inc files to discover the network, model the plan files, and insert the plan files into an archive repository. While this method can collect everything that can be collected through the Collector server or augmented method, unless one of the following conditions applies, it is recommended that you use either the Collector server or an augmented collection method for ease of maintainability.

- Multiple networks for use in the WAE Live application.
- SAM server (SAM_GETPLAN) integration.
- Other highly customized, advanced, or non-standard collection methods that require additional scripting or customized setups; this includes collection of different data at different frequencies.

To determine the best collection method for your purposes, see Collection by Configuration Method.

This chapter references the following terms.

- `$WAE_ROOT`—Location of the installation. By default, this is `/opt/cariden`.
- `$CARIDEN_HOME`—Directory in which the WAE Design, WAE Live, and WAE Collector executables and binaries are installed. The default is `/opt/cariden/software/mate/current`.

Note All instructions and examples assume you used `/opt/cariden` as the default installation directory. If you did not, then substitute your installation directory for `/opt/cariden`.

Workflow for Collecting Network Information Using Manual Collection

Before You Begin

We recommend that you back up all your configuration files.

Table 2-14 Manual Collection Configuration Workflow

Step	Task	Description/Notes
1	Perform Pre-Snapshot Configuration Tasks	Create an authentication file using the `mate_auth_init` tool, optionally edit the network access file, and create two sets of snapshot files for later use.
2	Modify Snapshot Files	Edit the snapshot files to customize collection, polling, modeling, etc. For examples of snapshot configurations, see Snapshot Examples.
3	Initialize Archive, Create Template, Run Collections	Post snapshot configuration.
4	Configure Continuous Polling and Collection Using Manual Collection	If using the data in applications, follow the steps described in this task.

Pre-Snapshot Configuration

Before editing and running the manual snapshots, you must do the following tasks:

Step 1 Run `mate_auth_init` to create an authentication file (`auth.enc`) used by SNMP and login tools.

```
mate_auth_init
```

This is an interactive tool that first prompts you to choose the SNMP version and the relevant parameters. For information, see Network Authentication.

Step 2 Optional: Customize network access. For information, see Network Access File.

Step 3 For new installations, copy the default snapshot.txt and snapshot.inc files to working configuration files.

```
cp /op. t/cariden/software/mate/current/etc/snapshot.txt /opt/cariden/etc
cp /opt/cariden/software/mate/current/etc/snapshot.inc /opt/cariden/etc
```

If this is not a new installation, you can use existing snapshot files in `/opt/cariden/etc`, and make modifications noted in this chapter as needed.

Modify Snapshot Files

For more information on how to modify snapshot files, see Snapshot Files.

Note A best practice is to add only a few tasks to the snapshot files, run the snapshot, and correct the errors. Then repeat this process until you have built the model of the network that you need.

Step 1 Edit the snapshot.txt file, which contains the collection, polling, modeling, and insertion tasks to perform. This file controls the sequence of execution and also contains environment variables of common values used in the snapshot.inc file.

- At minimum, you must define `unique`, `seed_router`, `igp`, and `home_dir`, `archive_dir`, and preferably the `backup_router`.

  By default, the `archive_insert` tool uses the `archive_dir` environment variables when inserting plan files into an external archive. Best practice is to use the default.

  Example: `archive_dir $(home_dir)/archives`

  To manually insert plan files into the WAE Live Map archive, create a new environment variable to specify the archive. Note that the location of the external archive and the Map archive must be different.

  Example: `map_archive_dir $(home_dir)/data/mldata`

- If needed, edit the `include` environment variable to read the snapshot.inc file from `$(home_dir)/etc`.

  ```
  include $(home_dir)/etc/snapshot.inc
  ```

Step 2 Define which tasks to execute to discover the network. Use the comments to enable or disable existing tasks, and add new tasks if needed. For examples, see Snapshot Examples.

For instructions specific to collecting flow data or SAM data, see Advanced Collection Configurations.

If you are discovering IS-IS, do the following:

- Uncomment the LOGIN_FIND_IGP_DB task, which discovers a basic IGP topology by logging into the seed router and parsing an IS-IS database. (To uncomment a task, remove the # sign.)
- Add a comment (#) to the beginning of the SNMP_FIND_OSPF_DB task.

  Example:

  ```
  <DISCOVERY_TASKS>
  #SAM_GETPLAN
  #SNMP_FIND_OSPF_DB
  ```

```
LOGIN_FIND_IGP_DB
SNMP_FIND_NODES
SNMP_FIND_INTERFACES
#GET_CONFIGS
#PARSE_CONFIGS
#FIND_BGP
SNMP_FIND_RSVP
SNMP_FIND_VPN
```

Step 3 Define which tasks to use for polling traffic.

Example:

```
<POLLING_TASKS>
SNMP_POLL
#POLL_LDP
```

Step 4 Define which tasks to execute to model the plan file. Use the comments to enable or disable existing tasks, and add new tasks if needed. If not using WAE Live, at minimum, uncomment COPY_FROM_TEMPLATE.

Example:

```
<ANALYSIS_TASKS>
#BUILD_MODEL
MATE_CONVERT
COPY_FROM_TEMPLATE
DMD_MESH_CREATOR
DMD_DEDUCT
#MATE_SIM
```

Step 5 Define which tasks to insert plan files. Use the comments to enable or disable existing tasks, and add new tasks if needed. For examples, see Snapshot Examples.

- ARCHIVE_INSERT—Insert the completed plan file into an external plan file archive that can be accessed by all the applications.
- ML_INSERT—Manually insert data into the WAE Live data store.
- MAP_ARCHIVE_INSERT—Manually insert plan files into the Map archive. Use only if using ML_INSERT and only if using the Map component. You must manually add this to the snapshot.inc file.

Example:

```
<ARCHIVE_INSERT_TASKS>
ARCHIVE_INSERT
ML_INSERT
MAP_ARCHIVE_INSERT
```

Step 6 Sometimes the IP management addresses that are discovered from the devices are different than the IP management addresses that are needed to communicate with the routers. If so, you need to create a <Nodes> table that lists the proper IP management addresses, and then use the `tab_merger` tool to insert the IP management addresses during the snapshot process. For information, contact your Cisco representative.

Step 7 As needed, edit the snapshot.inc file to modify and add tools that are to be called from the snapshot.txt file. You must add a definition for MAP_ARCHIVE_INSERT if you added that task.

Initialize Archive, Create Template, Run Collections

Note Text in <angle brackets> refers to environment variables that you set in the snapshot.txt file.

Step 1 Run `archive_init` to initialize the archive repository into which the plan files will be inserted. If you are using `archive_insert` to manually insert plan files into the WAE Live Map archive, this is not a required step.

```
archive_init -archive $WAE_ROOT/archives/<unique>-archive
```

Step 2 If collecting data for WAE Design Archive, use the `archive_config` tool to add the archive repository. At the same time, set up the template directory and template name.

```
archive_config -action add -name <unique> -path $WAE_ROOT/archives/<unique>-archive
-template-dir $WAE_ROOT/data -template-name <unique>-template.pln
```

Step 3 Create an empty template. You can ignore the warnings because the resulting file is an empty template file.

```
echo | mate_convert -plan-file - -out-file $WAE_ROOT/data/<unique>-template.pln
```

Note that WAE Live automatically creates the template.pln from the most recently collected plan file if no template exists. Therefore, for WAE Live, this step is not a requirement.

Step 4 Test the snapshot process by running it as a single tool to collect network data. Check the output for errors, fix them if needed, and rerun this test until it is successful before proceeding.

```
snapshot -config-file $WAE_ROOT/etc/snapshot.txt
```

Step 5 Create a cron job that repeats the process of creating snapshots and inserting them into the appropriate archive repository.

Note Both CARIDEN_ROOT and CARIDEN_HOME variables must be defined from within the crontab. You cannot use CARIDEN_HOME=$CARIDEN_ROOT/software/mate/current.

Open the file as follows.

```
crontab -e
```

At the end of the file, add the following lines.

```
CARIDEN_ROOT=/opt/cariden
CARIDEN_HOME=/opt/cariden/software/mate/current
0,15,30,45 * * * * $CARIDEN_HOME/bin/snapshot -config-file $CARIDEN_ROOT/etc/snapshot.txt
2>&1
```

Configure Continuous Polling and Collection Using Manual Collection

The manual collection method uses snapshot.txt and snapshot.inc files to discover the network, model the plan files, and insert the plan files into an archive repository. Optionally, it can push discovered topology to the WAE Network Interface (NI) server (using `collector_pushplan`), which can then

continuously poll traffic statistics and/or continuously discover PCEP LSPs; thereafter, an augmented snapshot can retrieve that plan file from the WAE NI server for further processing (using `collector_getplan`).

If configured, WAE Collector continuously collects LSPs managed by WAE. It also continuously polls LSP and interface statistics that are made available via SNMP.

This chapter references the following terms.

- `$CARIDEN_ROOT`—Location of the installation. By default, this is `/opt/cariden`.
- `$CARIDEN_HOME`—Directory in which the WAE Design, WAE Live, and WAE Collector executables and binaries are installed. The default is `/opt/cariden/software/mate/current`.

Note

- All instructions and examples assume you used `/opt/cariden` as the default installation directory. If you did not, then substitute your installation directory for `/opt/cariden`.
- You cannot use `sam_getplan` when using the WAE NI server.
- This chapter describes the full process of both pushing plan files to and retrieving them from the WAE NI server.

If you are using `collector_getplan` in an augmented snapshot after having configured the Collector server to push plan files to the WAE NI server, see Collecting Information Using Augmented Collection.

Workflow for Configuring Continuous Polling and Collection Using Manual Collection

Step 1 Best practice: Back up all configuration files before you begin.

Step 2 Set up the WAE NI server.

a. Configure Continuous Collection Parameters on the WAE NI Server.

b. Configure Authentication and Start Server.

Step 3 Execute Pre-Snapshot Configuration steps, which include creating an authentication file, optionally editing the network access file, and creating two sets of snapshot files for later use.

Step 4 Create Snapshot to Push Plan Files.

a. Configure Push Credentials.

b. Modify Push snapshot.txt to run only discovery tasks.

c. Modify Push snapshot.inc to include `collector_pushplan`.

Note If you do not need to run further tasks, such as creating demand meshes and running Demand Deduction, skip to Step 7.

Step 5 Create Snapshot to Get Plan Files.

a. Configure Get Credentials.

b. Configure Get Credentials to run post-discovery tasks.

c. Modify Get snapshot.inc to use `collector_getplan`.

d. Initialize Archive and Create Template.

Step 6 Run Collections.

Step 7 If using the data in applications, execute the Collecting Hardware Inventory steps.

Configure Continuous Collection Parameters on the WAE NI Server

Edit the `$WAE_HOME/wae-ni/etc/collection.cfg` file to tell the WAE NI server whether to continuously discover LSPs, as well as what to poll, how frequently to poll, and the amount of time to use when averaging the statistics.

Note Do not edit this file if you are collecting network information using the WAE Collector UI.

Parameter	Description
enablePcepLspCollection	True = Continuously collect PCEP LSPs. False = Do not continuously collect PCEP LSPs.

Continuously Poll Traffic Field Descriptions

Note Continuous polling applies to interfaces and LSP statistics that are made available via SNMP.

Parameter	Description
enableInterfaceStatsCollection	True = Continuously poll interface traffic. False = Do not poll interfaces.
interfaceStatsCollectionPeriodInSecs	The intervals (in seconds) between successive interface traffic counter polls. The minimum value is 60 seconds.
enableLspStatsCollection	True = Continuously poll LSP traffic. False = Do not poll LSPs.
lspStatsCollectionPeriodInSecs	The intervals (in seconds) between successive LSP traffic counter polls. The minimum value is 60 seconds.
enableQosStatsCollection	True = Continuously poll interface queue traffic. False = Do not poll interface queues.
enableVpnStatsCollection	True = Continuously poll VPN traffic. False = Do not poll VPNs.
statsComputingMinimumWindowLengthInSecs	This defines the minimum amount of time, in seconds, over which to generate averages of the polled traffic statistics. For example, if set to 300, to determine the rate of incoming packet errors, the WAE NI server takes the average of these incoming packet errors over the last 300 seconds. These traffic statistics are added to the plan file each time it is generated. The minimum value is 300 seconds.

statsComputingMaximumWindowLengthInSecs	There are times in which average statistics cannot be calculated. For instance, router response might be slow enough that the WAE NI server cannot get sufficient data. This parameter creates a safety net for such instances by giving the WAE NI server more time from which to collect data. The value is the percentage by which to expand (add to) the amount of time set in the statsComputingMinimumWindowLengthInSecs parameter if no statistics are collected. The lapses in statistics collection do not have to be synchronous for this parameter to apply. **Example:** If the statsComputingMinimumWindowLengthInSecs is 400 seconds and the statsComputingMaximumWindowLengthInSecs parameter is set to 25%, the window for calculating averages can be expanded up to 100 seconds (25% of 400 seconds) in the event no statistics are available at any time during the 10-minute window.
rawCounterTtlInMins	Defines the amount of time raw counters are kept in minutes. The minimum value is 5 minutes.
lspDiscoveryCollectionPeriodInSecs	Sets the LSP collection period in seconds (minimum is 60 seconds). This setting indicates how often the continuous poller will try to do LSP discovery.
lspDiscoveryUseCalculatedHops	Specifies whether to store calculated hops or actual hops in the plan file. The continuous poller collects both during discovery. True—Stores calculated hops, if available, in the plan file. False—Stores actual hops in the plan file.
lspDiscoveryUsePcepSignaledName	Specifies whether to use the PCEP signaled name while storing the LSP in the plan file. True—PCEP signaled name is used as the LSP name in plan file. False—The LSP name that is set on the router is used as the name in the plan file. **Note** If set to true, the Cisco router LSP name is stored as tunnel-te5.
lspDiscoveryUseAutobandwidth	Specifies whether or not to store the auto bandwidth in the plan file. True—If the auto bandwidth rate is discovered, then auto bandwidth is stored as the setup bandwidth in the plan file. False—The discovered setup bandwidth is stored in plan file.
l1DiscoveryEnabled	True—Enables Layer 1 discovery in WAE NI. False—Disables Layer 1 discovery in WAE NI.

Common Parameters

Parameter	Description
logVerbosity	Integer that defines the verbosity of the information returned by log files. Trace = 60 Debug = 50 Info = 40 Warn = 30 Error = 20
planFileGenerationIntervalInSecs	Defines how often a pre-calculated plan file is generated in seconds. The minimum value is 300 seconds. This value is used when the WAE NI server is configured through the WAE Collector UI. Note that both `collector_getplan` and WAE Live use on-demand plan files rather than pre-calculated plan files.

Configure Authentication and Start Server

Step 1 Configure the authentication for the WAE NI server.

default username: admin

default password: cariden

Step 2 If it is not running, start the WAE NI server.

Check the status: `service wae-ni status`

Start: `service wae-ni start`

Pre-Snapshot Configuration

Note For demonstration purposes, this chapter references two sets of files: snapshot-pushplan and snapshot-getplan .txt and .inc files. You can name these files whatever you choose. Therefore, where text states, for example, snapshot-pushplan.txt, this means the snapshot.txt file that is pushing the plan file to the WAE NI server. Additionally, all instructions and examples assume you used `/opt/cariden` as the default installation directory. If you did not, then substitute your installation directory for `/opt/cariden`.

Step 1 Run `mate_auth_init` to create an authentication file (`auth.enc`) used by SNMP and login tools.

```
mate_auth_init
```

This is an interactive tool that first prompts you to choose the SNMP version and the relevant parameters. For information, see Network Authentication.

Step 2 Optional: Customize network access. see Network Access File.

Step 3 Because you need to run two snapshots, create both sets of files now. Later you will edit both sets of files.

Copy the default snapshot.txt, snapshot.inc, snapshot_augment_collector.txt and snapshot_augment_collector.inc files to working configuration files in `$WAE_ROOT/etc` and **give them different names**.

Note that if you have existing snapshot files in `$WAE_ROOT/etc`, you can copy those files to snapshot-pushplan and snapshot-getplan files, and then make changes to those files aligned with the instructions in this chapter.

Examples:

```
cp /opt/cariden/software/mate/current/etc/snapshot.txt
/opt/cariden/etc/snapshot-pushplan.txt
cp /opt/cariden/software/mate/current/etc/snapshot.inc
/opt/cariden/etc/snapshot-pushplan.inc
cp /opt/cariden/software/mate/current/etc/snapshot_augment_collector.txt
/opt/cariden/etc/snapshot-getplan.txt
cp /opt/cariden/software/mate/current/etc/snapshot_augment_collector.inc
/opt/cariden/etc/snapshot-getplan.inc
```

Create Snapshot to Push Plan Files

Note For information on configuring snapshot.txt and snapshot.inc files, see Snapshot Files. This section assumes you know how to modify these files and how they work together.

Configure Push Credentials

Run the `collector_pushplan` tool once to set the WAE NI server's credentials for later use in the snapshot files. You must use `-set-credentials true`. When prompted, enter the username and password for the WAE NI server.

The default credential file is `$CARIDEN_ROOT/etc/collector/credentials.enc`. **The credentials file for the snapshot-pushplan and the snapshot-getplan files must be the same.** To change it, use the `-credentials-file` option.

Example: `collector_pushplan -set-credentials true -credentials-file /opt/cariden/etc/collector/credentials-CP.enc`

Note The credentials file used for the Collector server and WAE NI server must be different.

Modify Push snapshot.txt

Step 1 Define the environment variables in the <ENVIRONMENT> section. Each parameter must be separated from its value by a TAB.

- At minimum, define `unique`, `seed_router`, `igp`, and `home_dir`, and preferably the `backup_router`.
- If needed, edit the `include` environment variable to read the snapshot-pushplan.inc file from `$(home_dir)/etc`.

  Example: `include $(home_dir)/etc/snapshot-pushplan.inc`

Step 2 In the <DISCOVERY_TASKS> section, uncomment or add tasks that discover the topology. (See example in step 3.) Note that the order of these tasks determines the sequence of their execution.

Step 3 Immediately following the discovery tasks, add a COLLECTOR_PUSHPLAN task to push the plan file to the WAE NI server.

Example:

```
<DISCOVERY_TASKS>
#SAM_GETPLAN
SNMP_FIND_OSPF_DB
#LOGIN_FIND_IGP_DB
SNMP_FIND_NODES
SNMP_FIND_INTERFACES
#GET_CONFIGS
#PARSE_CONFIGS
#FIND_BGP
SNMP_FIND_RSVP
#SNMP_FIND_VPN
COLLECTOR_PUSHPLAN
```

Step 4 **Either remove or comment out all other tasks in the snapshot**.

Step 5 Sometimes the IP management addresses that are discovered from the devices are different than the IP management addresses that are needed to communicate with the routers. If so, you need to create a <Nodes> table that lists the proper IP management addresses, and then use the `tab_merger` tool to insert the IP management addresses during the snapshot process. For information, contact your Cisco representative.

Modify Push snapshot.inc

Step 1 As needed, edit and add collection tools that are to be called from the snapshot-pushplan.txt file.

Step 2 Add the `collector_pushplan` configuration. The required options are `-set-credentials`, `-credentials-file`, `-in-net-access-file`, and `-in-auth-file`.

Set the `-set-credentials` to `false` so that the snapshot process does not stop to ask for credentials. You only need to set this to `true` once, which you have already done (as per Configure Push Credentials).

The `-credentials-file` must match the name that you specified when you first set the credentials (as per Configure Push Credentials).

The `-in-plan-file` tells the WAE NI server the path and name of the plan file that is being sent to it.

By default, the net_access.txt file is in `$CARIDEN_HOME/etc`. If you modify this, then that same path and name must be configured for `collector_pushplan`, and it must reside in one of these locations.

- `~/.cariden/etc`
- `$CARIDEN_ROOT/etc`
- `$CARIDEN_HOME/etc`

The auth.enc file location must match the location in which the `mate_auth_init` put it.

JMS is the protocol that the Collector server uses to communicate with the WAE NI server. By default, the WAE NI server is using the same host (localhost) as the Collector server. By default, the WAE NI server listens on port 61617 to receive plan files pushed to it. You can change these using the `-jms-server-address` and `-jms-server-port` options.

Example COLLECTOR_PUSHPLAN

Name	Required Value
<COLLECTOR_PUSHPLAN>	
cmd	$(cariden_home)/bin/collector_pushplan
cmd_opt	COLLECTOR_PUSHPLAN_CMD_OPT
postcmd	cp
postcmd_opt	COLLECTOR_PUSHPLAN_CP_CMD_OPT
cmd_success	0
<COLLECTOR_PUSHPLAN_CMD_OPT>	
set-credentials	false
credentials-file	$(home_dir)/etc/collector/credentials-CP.enc
in-plan-file	$(work_dir)/$(unique).txt
in-net-access-file	$(cariden_home)/etc/net_access.txt
in-auth-file	$(home_dir)/etc/auth.enc
jms-server-address	localhost
jms-server-port	61617
<COLLECTOR_PUSHPLAN_CP_CMD_OPT>	
	$(work_dir)/$(unique).txt
	$(debug_dir)/$(unique).txt-post-collector_pushplan.txt

Create Snapshot to Get Plan Files

Note If you do not need to add further tasks to the snapshots, skip this section and go to Run Collections.

Configure Get Credentials

Run the `collector_getplan` tool once to set the WAE NI server's credentials for later use in the snapshot files. You must use `-set-credentials true`.

The default credential file is `$CARIDEN_ROOT/etc/collector/credentials.enc`. **The credentials file for the snapshot-pushplan and the snapshot-getplan files must be the same.** To change it, use the `-credentials-file` option.

Example: `collector_getplan -set-credentials true -credentials-file /opt/cariden/etc/collector/credentials-CP.enc`

Note The credentials file used for the Collector server and WAE NI server must be different.

Modify Get snapshot.txt

The instructions in this chapter use the `archive_insert` tool to insert plan files into an external archive. For information on manually inserting plan files into WAE Live, see Snapshot Examples.

Step 1 Define the environment variables in the <ENVIRONMENT> section. Each parameter must be separated from its value by a TAB.

- At minimum, define `unique`, `seed_router`, `igp`, and `home_dir`, and preferably the `backup_router`. **These must be the same as in the snapshot-pushplan.txt file.** The `archive_dir` must also be specified, and it is not relevant to the snapshot-pushplan.txt file.
- Add or edit the `collector_url` variable to set to the location of the WAE NI server. The default port on which it listens for incoming plans is 8086.

  Example: `collector_url https://localhost:8086`
- If needed, edit the `include` environment variable to read the snapshot-getplan.inc file from `$(home_dir)/etc`.

  Example: `include $(home_dir)/etc/snapshot-getplan.inc`

Step 2 Keep COLLECTOR_GETPLAN uncommented as the first task. Either remove or comment out all **tasks used in discovering the topology or polling for traffic.**

Example:

```
<DISCOVERY_TASKS>
COLLECTOR_GETPLAN
#GET_CONFIGS
#PARSE_CONFIGS
#SNMP_FIND_VPN
<POLLING_TASKS>
#SNMP_POLL
#POLL_LDP
```

Step 3 Define whether to execute flow collection, define which tasks to execute to model the plan file, and define an insert task to specify where to insert the final plan files. Use the comments to enable or disable existing tasks, and add new tasks if needed. Note that the order of these tasks determines the sequence of their execution. At minimum, uncomment the following tasks.

- COPY_FROM_TEMPLATE—Copies selected values from the template plan file into the newly generated plan, while preserving network configuration information.
- ARCHIVE_INSERT—Stores the completed plan file in an external plan file archive for use by applications.

Example:

```
<FLOW_TASKS>
FLOW_GET
<ANALYSIS_TASKS>
#BUILD_MODEL
MATE_CONVERT
COPY_FROM_TEMPLATE
DMD_MESH_CREATOR
DMD_DEDUCT
#MATE_SIM
<ARCHIVE_INSERT_TASKS>
ARCHIVE_INSERT
#ML_INSERT
```

Modify Get snapshot.inc

Step 1 As needed, add or edit flow, modeling, and insertion tools that are to be called from the snapshot-getplan.txt file.

Step 2 Keep the `collector_getplan` configuration `-url` option set to the `collector_url` environment variable.

Keep `-set-credentials` to `false` so that the snapshot process does not stop to ask for credentials. You only need to set this to `true` once, which you have already done (as per Configure Get Credentials).

The `-credentials-file` must match the name that you specified when you first set the credentials (as per Configure Get Credentials), and it must be the same as used in the snapshot-pushplan.inc file.

The `-out-file` tells the WAE NI server where (path and filename) to write the latest plan file.

The net_access_session_file.txt and auth_session_file.enc must reside in one of the following locations. Best practice is to put them wherever you put the net_access.txt and auth.enc file used in the snapshot-pushplan.inc file.

- `~/.cariden/etc`
- `$CARIDEN_ROOT/etc`
- `$CARIDEN_HOME/etc`

Example COLLECTOR_GETPLAN

Name	Required Value
<COLLECTOR_GETPLAN>	
cmd	$(cariden_home)/bin/collector_getplan
cmd_opt	COLLECTOR_GETPLAN_CMD_OPT
postcmd	cp
postcmd_opt	COLLECTOR_GETPLAN_CP_CMD_OPT
cmd_success	0
<COLLECTOR_GETPLAN_CMD_OPT>	
set-credentials	false
credentials-file	$(home_dir)/etc/collector/credentials-CP.enc
get	files
url	$(collector_url)
if-later-than-timestamp-file	$(timestamp_file)
out-file	$(work_dir)/$(unique).txt
out-net-access-file	$(net_access_session_file)
out-auth-file	$(auth_session_file)
<COLLECTOR_GETPLAN_CP_CMD_OPT>	
	$(work_dir)/$(unique).txt
	$(debug_dir)/$(unique).txt-post-collector_getplan.txt

Initialize Archive and Create Template

Note Text in <angle brackets> refers to environment variables that you set in the snapshot-getplan.txt file.

Step 1 Run `archive_init` to initialize the archive repository into which the plan files will be inserted.

```
archive_init -archive $WAE_ROOT/archives/<unique>-archive
```

Step 2 If collecting data for WAE Design Archive, use the `archive_config` tool to add the archive repository. At the same time, set up the template directory and template name.

```
archive_config -action add -name <unique> -path $WAE_ROOT/archives/<unique>-archive
-template-dir $WAE_ROOT/data -template-name <unique>-template.pln
```

Step 3 Create an empty template. You can ignore the warnings because the resulting file is an empty template file.

```
echo | mate_convert -plan-file - -out-file $WAE_ROOT/data/<unique>-template.pln
```

Note that WAE Live automatically creates the template.pln from the most recently collected plan file if no template exists. Therefore, for WAE Live, this step is not required.

Run Collections

Step 1 Test the snapshot process by running each one as a single tool to collect network data. Check the output for errors, fix them if needed, and rerun this test until it is successful before proceeding.

```
snapshot -config-file $WAE_ROOT/etc/snapshot-pushplan.txt
snapshot -config-file $WAE_ROOT/etc/snapshot-getplan.txt
```

Step 2 Create a cron job that repeats the process of creating snapshots and inserting them into the archive repository.

Note Both CARIDEN_ROOT and CARIDEN_HOME variables must be defined from within the crontab. You cannot use CARIDEN_HOME=$CARIDEN_ROOT/software/mate/current.

Open the file for editing as follows.

```
crontab -e
```

At the end of the file, add the following lines. If you used only the snapshot-pushplan configuration, do not add snapshot-getplan to the cron job.

```
CARIDEN_ROOT=/opt/cariden
CARIDEN_HOME=/opt/cariden/software/mate/current
SNAPSHOT="/opt/cariden/software/mate/current/bin/snapshot -log-to-screen false"
*/30 * * * * $SNAPSHOT -config-file $CARIDEN_ROOT/etc/snapshot-pushplan.txt
*/30 * * * * $SNAPSHOT -config-file $CARIDEN_ROOT/etc/snapshot-getplan.txt
```

Collecting Hardware Inventory

To easily collect and view hardware inventory information on your network, run the snapshot tool using the snapshot_hardware_inventory.txt file, and then view the information in WAE Live. For more information on WAE Live, see the *WAE LIVE User Guide*.

Prerequisite

- Run the following command:

  ```
  collector_getplan -set-credentials true
  ```

- A collection has been done using one of the collection methods and a plan file exists.
- `$CARIDEN_ROOT`—Location of the installation. By default, this is `/opt/cariden`. All instructions and examples assume `/opt/cariden` as the default installation directory. If you did not use the default, substitute your installation directory for /opt/cariden
- `$CARIDEN_HOME`—Directory in which the WAE Design, WAE Live, and WAE Collector executables and binaries are installed. The default is `/opt/cariden/software/mate/current`.

Step 1 Copy the snapshot_inventory_inc and snapshot_inventory.txt files from $CARIDEN_HOME/etc to $CARIDEN_ROOT/etc.

```
cp $CARIDEN_HOME/etc/snapshot_hardware_inventory* $CARIDEN_ROOT/etc
```

Step 2 To collect hardware inventory, enter the following command:

```
snapshot -config-file $CARIDEN_ROOT/etc/snapshot_hardware_inventory.txt
```

Note You might see errors because of third-party devices. You can ignore these errors.

Step 3 Check the output for errors, fix them if needed, and rerun this test until it is successful before proceeding.

Step 4 Create a cron job that repeats the process of creating snapshots to collect hardware inventory and inserting them into the WAE Live data store once a day.

Both `CARIDEN_ROOT` and `CARIDEN_HOME` variables must be defined from within the crontab. We recommend that you place these definitions at the top of the crontab because they are used globally within multiple crontab commands.

You cannot use `CARIDEN_HOME=$CARIDEN_ROOT/software/mate/current`. For example, you cannot use `$CARIDEN_HOME/bin/snapshot -config-file $CARIDEN_ROOT/etc/snapshot_hardware_inventory.txt` unless `$CARIDEN_HOME` and `$CARIDEN_ROOT` have been previously defined within crontab.

Open the file as follows.

```
crontab -e
```

At the end of the file, add the following lines:

```
CARIDEN_ROOT=/opt/cariden
CARIDEN_HOME=/opt/cariden/software/mate/current
0 0 * * * $CARIDEN_HOME/bin/snapshot -config-file $CARIDEN_ROOT/etc/snapshot_inventory.txt
2>&1
```

Customizing and Understanding Hardware Inventory Collection

Inventory collection collects and processes network hardware information to create the NetIntNodeInventory table used by WAE Live to produce inventory reports.

The following table lists the tasks that are performed in the snapshot_hardware_inventory.txt file and some of the options that can be edited in the snapshot_hardware_inventory.inc file.

Task	Description/Notes
COLLECTOR_GETPLAN	Calls the plan file.
GET_INVENTORY	Collects the network hardware and creates NetIntHardware tables that contain every device collected from MIB walks segregated by object type. The `get_inventory` tool also uses SSH and NETCONF to collect data that is not available in MIBs. To allow logging in to the router to collect inventory data, you can set the `get_inventory -login-allowed` option to `true`. By default, it is set to `true`.
BUILD_INVENTORY	Processes the raw hardware data information in the NetIntHardware* tables) to categorize and remove unwanted objects in the final NetIntNodeInventory table. To broaden the search when processing raw inventory data, you can set the `build_inventory -guess-template-if-nomatch` option to `true`.
MATE_CONVERT	Converts the plan .txt file to a .pln file
ML_INSERT_CTL	Inserts and schedules the insertion of inventory data into the WAE Inventory data store.

Collected Hardware

The `get_inventory` tool creates a series of NetIntHardware* tables that store the collected hardware information based on hardware type. While these tables are not directly usable by WAE Live, four of them are processed by `build_inventory` for use in WAE Live. Each of the following objects are defined by node IP address and SNMP ID.

- NetIntHardwareChassis—Router chassis objects identified by node IP address and SNMP ID.
- NetIntHardwareContainer—Each entry represents a slot in a router (anything that can have a field replaceable unit (FRU) type device installed into it). Examples include chassis slots, module slots, and port slots.
- NetIntHardwareModule—Hardware devices that can be installed into other hardware devices. Generally, these devices directly support traffic such as linecards, modules, and route processors, and do not fall into one of the other function-specific hardware tables.
- NetIntHardwarePort—Physical ports on the router.

Hardware Hierarchy

The hardware has a parent-child relationship based on where the object resides within the router. The chassis has no parent and is considered the *root object*. Other than the chassis, each object has one parent and can have one or more child objects. Objects with no children are called *leaf objects*, such as ports and empty containers. This hierarchy generally reflects how hardware objects are installed within other objects. For instance, a module representing a linecard might have a parent object that is a container representing a slot.

The parent is identifiable in the NetIntHardware* tables by the ParentTable and ParentId columns. Using these two columns along with the Node (node IP address) column, you can find the parent object for any hardware object.

Example: This NetIntHardwareContainer entry identifies that container 172.23.123.456 has a chassis as a parent. In the NetIntHardwareChassis, there is an SnmpID entry that matches the container's ParentId of 2512347.

NetIntHardwareContainer

Node	SnmpID	ParentID	Model	Name	NumChildren	ParentTable	SlotNumber
172.23.123.456	2503733	2512347		slot mau 0/0/0/5	0	NetIntHardwareChassis	0

Tracing the hierarchy from each leaf object to its corresponding root object based on the parent-child relationships results in a series of object types that form its hardware hierarchy. It is this trace that the `build_inventory` tool uses to determine how to process the hardware devices. This is also the process you must use if adding an entry to the HWInventoryTemplates table.

Example: Chassis-Container-Module-Module-Container-Port

Tables for Processing Inventory

The `build_inventory` tool constructs the NetIntNodeInventory table by processing the NetIntHardware* tables. The tool requires two configuration files and can additionally use an optional one. If not specified, the files included in the `$CARIDEN_HOME/etc/inventory` are used.

- master_inventory_templates.txt (required)—This file contains these tables.
  - HWInventoryTemplates entries categorize the devices in the final NetIntNodeInventory table, as well as prune from inclusion.
  - HWNameFormatRules entries format hardware object names to make them more usable, as well as correct unexpected SNMP results.
- master_exclude_list.txt (required)—Contains the ExcludeHWList table that prevents (blacklists) hardware objects from being included in the final NetIntNodeInventory table. This can be useful when for excluding hardware that does not forward or carry traffic.
- master_hw_spec.txt (optional)—Contains the HardwareSpec table that can be used to adjust collected data in terms of the number of slots in a specified device when the slots returned by SNMP is inaccurate.

If you modify the template or choose to exclude files, you will want these changes to persist across software upgrades. To do so, you must move these files from $CARIDEN_HOME to $CARIDEN_ROOT and update the snapshot files accordingly.

1. Copy $CARIDEN_HOME/etc/inventory to $CARIDEN_ROOT/etc:

   ```
   cp -r $CARIDEN_HOME/etc/inventory $CARIDEN_ROOT/etc
   ```

2. Copy $CARIDEN_HOME/etc/snapshot_hardware_inventory.txt and .inc to $CARIDEN_ROOT/etc/inventory:

   ```
   cp $CARIDEN_HOME/etc/snapshot_hardware_inventory.*
   $CARIDEN_ROOT/etc/inventory/
   ```

3. Run the snapshot manually:

   ```
   $CARIDEN_HOME/bin/snapshot -config-file
   ```

```
$CARIDEN_ROOT/etc/inventory/snapshot_hardware_inventory.txt
```

4. Schedule the snapshot in crontab:

```
0 0 * * * $CARIDEN_HOME/bin/snapshot -config-file
$CARIDEN_ROOT/etc/inventory/snapshot_hardware_inventory.txt 2>$1t
```

Configure Hardware Templates

The `build_inventory -template-file` option calls a file containing both the HWInventoryTemplates and the HWNameFormatRules tables, which by default are in the `$CARIDEN_HOME/etc/inventory/master_inventory_templates.txt` file.

HWInventoryTemplates Table

The HWInventoryTemplates table tells the `build_inventory` tool how to interpret hardware referenced by the NetIntHardware* tables. It enables `build_inventory` to categorize objects into common, vendor-neutral hardware types, such as chassis, linecards, and slots, as well as to remove hardware types that are not of interest.

Inventory hardware is categorized as a chassis, slot, linecard, module slot, module, port slot, port, or transceiver. A container is categorized as either a slot, module slot, or port slot. A module is categorized as either a module or a linecard. All other hardware objects are categorized by their same name. For instance, a chassis is categorized as a chassis. These categorized hardware objects are available through the WAE Live application for use in inventory reports.

The build_inventory tool looks at the following columns of the HWInventoryTemplates table for matches in the NetIntHardware* tables in this order.

- DiscoveredHWHierarchy, Vendor, Model
- DiscoveredHWHierarchy, Vendor, * (where * means all entries in the Model column)

You can further enhance the search using the `-guess-template-if-nomatch true` option. In this instance, if no matches are found using the first two criteria, WAE Collector then looks for matches only for DiscoveredHWHierarchy and Vendor, and does not consider Model.

If a match is found, the subsequent columns after DiscoveredHWHierarchy tell `build_inventory` how to categorize the hardware. These latter columns identify hardware object types: chassis, slot, linecard, module slot, module, port slot, port, or transceiver. Each column entry has the following format. For an example, see Figure 2-4.

Type,Identifier,Name

- Type is the discovered hardware type, such as “container.”
- Identifier specifies which object (of one or more of the same type) in the hierarchy is referenced (0, 1, ...).
- Name specifies a column heading in the NetIntHardware* table. This is the name that appears in for that object in the NetIntNodeInventory table and thus, in WAE Live inventory reports.

Example: Module,0,Model

(Model is a column heading in the NetIntHardwareModule table)

Multiple name source columns can be specified with a colon.

Example: Container,0,Model:Name

If a hardware category does not exist or is empty, `build_inventory` does not include it in the final NetIntNodeInventory table.

Example

Using the first row of the default master_inventory_templates.txt file, WAE Collector searches the NetIntHardware* tables for ones that have entries that match the Vendor, Model, and DiscoveredHWHierarchy columns, as follows.

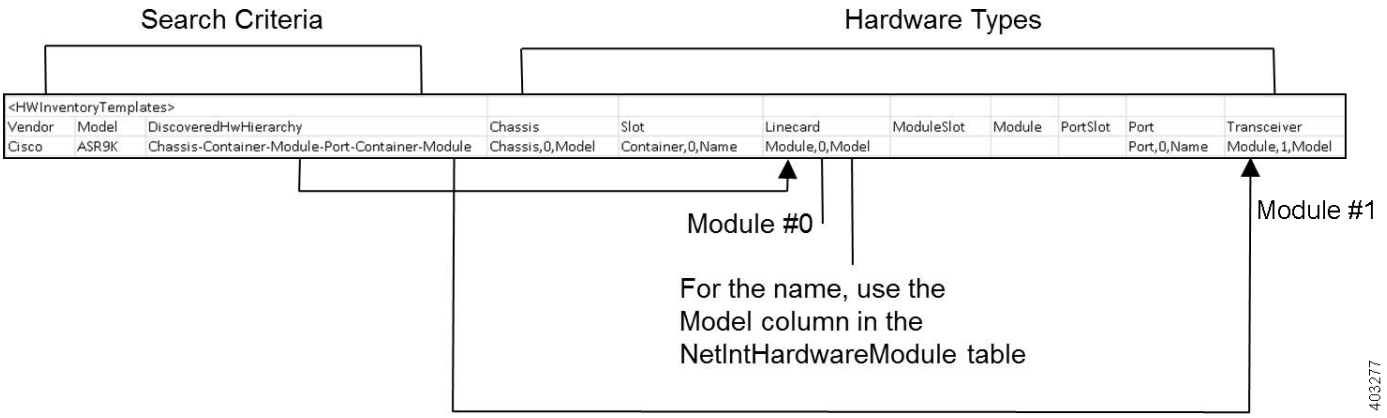
Cisco ASR9K Chassis-Container-Module-Port-Container-Module

Thereafter, it categorizes each entry in the hardware hierarchy (DiscoveredHWHierarchy column), and defines its location in the hardware types columns.

The first Module entry is defined as a linecard, it is identified as #0, and the name that appears in the NetIntNodeInventory table is the one appearing in the Model column of the NetIntHardwareModule table. The second module is defined as a transceiver object and is identified as #1. It uses the same name format.

Notice that there are two containers in the hierarchy, but there is only one defined as a Type. This means that the second container would not appear in the NetIntNodeInventory table.

Figure 2-4 Example HWInventoryTemplates Entry



Add HWInventoryTemplates Entries

If WAE Collector encounters an inventory device that is not in the HWInventoryTemplates table, it generates a warning that specifies pieces of the hardware hierarchy, including the SNMP ID of the leaf object and the IP address of the router. You can use this information to manually trace the objects from the leaf to the root and derive an appropriate entry in the HWInventoryTemplates table. For information on tracing hardware hierarchies, see Hardware Hierarchy.

Step 1 Copy the warning message for reference, and use it for Step 2.

Step 2 Using the router's IP address, as well as the SNMP ID, name, and model of the leaf object, find the leaf object referenced in the warning in either the NetIntHardwarePort or the NetIntHardwareContainer table.

Step 3 Use the leaf object's ParentTable and ParentId columns to trace the leaf back to its parent. For each successive parent, use its ParentTable and ParentId columns until you reach the root object (chassis) in the NetIntHardwareChassis table.

Step 4 Once each object in the hardware hierarchy is found, add it to the DiscoveredHWHierarchy column of the HWInventoryTemplates table. Also complete the Vendor and Model columns.

Step 5 For each object in the hardware hierarchy (DiscoveredHWHierarchy column), classify it into one of the standard hardware types, which are the columns listed after the DiscoveredHWHierarchy column.

HWNameFormatRules Table

The HWNameFormatRules table specifies how to format the names in the NetIntNodeInventory table. This is useful for converting long or meaningless names to ones that are easier to read and clearer for users to understand.

For each entry in the HWInventoryTemplates table, the HWNameFormatRules table is searched for a matching vendor, hardware type (HWType), name (PatternMatchExpression). Then, rather than using the name specified in the HWInventoryTemplates table, the NetIntNodeInventory table is updated with the name identified in the ReplacementExpression column.

If multiple matches apply, the first match found is used. Both the PatternMatchExpression and the ReplacementExpression can be defined as a literal string in single quotes or as a regular expression.

Example: The entries in the table work as follows.

- Replaces all Cisco chassis name with 7507 if the name has four characters where A is the beginning of the string and Z is the end of the string.
- Replaces all Cisco linecard names that match 800-20017-.* with 1X10GE-LR-SC.
- Replaces all Juniper chassis named "Juniper (MX960) Internet Backbone Router" with MX960.

HWNameFormatRules			
Vendor	HWType	PatternMatchExpression	ReplacementExpression
Cisco	Chassis	\A4\Z	'7507'
Cisco	Linecard	800-20017-.*	'1X10GE-LR-SC'
Juniper	Chassis	Juniper (MX960) Internet Backbone Router	$1

Note SNMP returns many slot names as text, rather than integers. It is a best practice to remove all text from slot numbers for optimal use in WAE Live inventory reports.

Exclude Hardware by Model or Name

The `build_inventory -exclude-file` option calls a file containing the ExcludeHWList table, which by default is in the `$CARIDEN_HOME/etc/inventory/master_exclude_list.txt` file. This table enables you to identify hardware objects to exclude from the NetIntNodeInventory table based on model, name, or both. This is useful, for instance, when excluding management ports and route processors. The model and names can be specified using regular expressions or they can be literals.

Example: The entries in the table work as follows.

- Exclude all objects in the NetIntHardwarePort table where the vendor is Cisco and the name ends with CPU0/129.
- Exclude all objects in the NetIntHardwareModule table where the vendor is Cisco and the model is 800-12308-02.
- Exclude all objects in the NetIntHardwarePort table where the vendor is Cisco and the name is Mgmt.

ExcludeHWList			
HWTable	Vendor	Model	Name
NetIntHardwarePort	Cisco		\/CPU0\/129$
NetIntHardwareModule	Cisco	800-12308-02	
NetIntHardwarePort	Cisco		Mgmt

HardwareSpec

The `build_inventory -hardware-spec-file` option calls a file containing the HardwareSpec table, which by default is in the `$CARIDEN_HOME/etc/inventory/master_hw_spec.txt` file. This table enables you to adjust data returned from SNMP. You can adjust both the total number of slots (TotSlot) and the slot numbering range (SlotNum). For instance, SNMP might return 7 slots for a chassis when there are actually 9, including route processors.

This table looks only for hardware that contains slots, module slots, or port slots, and thus, the hardware type (HWType column) must be chassis, linecard, or module. SlotNum indicates the slot number range. For instance, some routers start with slot 0, whereas others start with slot 1.

Example: This table entry sets the Cisco 7609 chassis to have a total of 9 slots and to start the slot numbering with 9.

HardwareSpec				
Vendor	HWType	Model	TotSlot	SlotNum
Cisco	Chassis	7609	9	1-9

Troubleshooting Collection

When collecting network information using the WAE Collector UI, you can use the WAE Collector UI to check for node access failures, nodes that are not responding, or other problems with collecting data. Using this information, you can correct the problems, often by setting override rules for problematic nodes or changing the global rules for collecting data. For example, if nodes with SNMP community strings differ from the majority of the discovered nodes, you can individually configure them to use specific SNMP community strings. Once such changes are applied, they take effect for the next instance of data collection.

WAE Collector Server Logging

On the Schedule page, you can configure the Collector server to generate detailed log files that are viewable on both the Status and Log pages.

If you need to contact Cisco support, we recommend that you first run Download Diagnostics or `mate_tech_support`, and send the resulting file to your representative.

- On the Status page, use the Download Diagnostics feature to create a .zip file containing the state of the local Collector server during the last collection.
- The `mate_tech_support` tool creates .tar file containing information for the Collector server, WAE NI server, WAE Core server, and WAE Live. Note this tool is applicable only if all three servers are on the same local device. For information on `mate_tech_support`, refer to its `-help` output.

For all event logs of all servers in an HA environment, go to the WAE Statistics > Events page. For diagnostic and process status information for all servers, go to the WAE Statistics > Diagnostics and WAE Statistics > Processes page, respectively.

WAE NI Logging

The WAE NI log file is located in `$WAE_ROOT/logs/wae-ni/collector-core.log`.

To change the log level at runtime, edit `$WAE_HOME/wae-ni/etc org.ops4j.pax.loggin.cfg`. Edit the `log4j.ogger.com.cisco=<log_level>` parameter where <log_level> is the minimum severity level you want displayed. For example, if `log4j.ogger.com.cisco=DEBUG`, then all severity levels set to DEBUG or higher will be captured during runtime. The log level severities are listed in the following order (from highest to lowest): FATAL, ERROR, WARN, INFO, and DEBUG.

Viewing a Network Model

The network Model Manager provides an easy way to copy a template plan file into a newly generated plan file, store the resulting plan file into a plan file archive, and view the network model that includes the creation of demands after collection. Rather than using Augmented Collection (see Collecting Information Using Augmented Collection) to perform these tasks, the Model Manager provides this capability without having to configure these tasks outside the WAE UI.

Viewing a Network Model Workflow

Step 1 Complete one network collection.

Step 2 Create and Schedule the Model Manager.

Step 3 View status to see that it completes. See View Event Information.

Step 4 View the Network Model using WAE Design Archive.

Create and Schedule the Model Manager

Step 1 Schedule and complete at least one network collection. For more information, see "Workflow for Collecting Basic Information Using the WAE Collector UI" section on page 2-10.

Step 2 From the WAE UI, select **Model Manager** and click the add **(+)** icon.

Step 3 Enter a name for the network model.

Step 4 Enter how often (in minutes) you want the Model Manager to retrieve the plan file from the selected network collection.

Step 5 From the Select Network drop-down list, select the applicable network you want to view a model for. To create a merged network model of multiple network collections, click the add **(+)** icon and select additional networks.

Step 6 Click **Save**. The Model Manager list appears.

Step 7 From the Status column, click the toggle button so that the applicable network model is running. By default, the status is "Stopped" until you enable it to run.

View the Network Model

After the Model Manager retrieves the plan file, you can open it using WAE Design Archive to get a visual representation of the network.

Step 1 To open the network model, click **Archive**. The WAE Design Archive appears.

Step 2 From the applicable network model, click **Map**. For more information on WAE Design Archive, click **Help** or see the *Cisco WAE Design Archive User and Administration Guide*.

View Event Information

To view the snapshot log for a network model, click the **Log** icon from the Actions column.

To view only the events for the last snapshot, click the **Summary** icon.

Advanced Configuration - Modify Snapshot Tasks

Only advanced users who are familiar with snapshot tasks should edit the Advance Configuration page. To modify or add additional snapshot tasks, do the following:

Step 1 If Model Manager is running on the network, stop it. From the Status column, click the toggle button so that the applicable network model is stopped.

Step 2 Click the **Advance Configuration** (pencil) icon associated with the applicable network. The Advanced Configuration page for the selected network model appears.

Step 3 Modify the file.

Step 4 Click **Save**.

Step 5 To run the network model and validate your changes, click **Run Once**. A snapshot log for the selected network appears.

Advanced Collection Configurations

A comprehensive set of online and offline tools are available to discover and retrieve information from an operational network for input into a plan file. There are various methods available, depending on sources of information and network access, such as SNMP access and router configuration files, and what information is to be imported.

Advanced collection configuration includes the following topics:

- Multi-Network Collection—Describes how to collect data from multiple networks and insert them into either an external archive or directly into the WAE Live data store.
- Offline Discovery—Describes the tools used to discover and retrieve information from router configurations and from RRD tools.
- Network Access File—Describes how to customize network access files that store network access parameters, such as time-out and retry settings.
- Network Authentication—Describes how to configure the authentication file. The file keeps SNMP community and router login authentication information for use by WAE Collector.
- Manage Archives—Describes the basic archive tools that apply to both the WAE Live and WAE Design Archive applications when using an augmented or manual discovery method.

Terminology

This chapter uses the following terms.

- `$WAE_ROOT`—Location of the installation. By default, this is `/opt/cariden`. These terms are interchangeable.
- `$WAE_HOME`—Directory in which the WAE Design, WAE Live, and WAE Collector executables and binaries are installed. The default is `/opt/cariden/software/mate/current`.

Multi-Network Collection

Note This section describes running multi-network collections using the manual collection method. You can also use the WAE UI to run multi-network collections. For more information on using the WAE UI, see Add Additional Networks for Collection.

The manner in which you configure multiple networks for inclusion in WAE Live depends on whether you are using an external archive or directly inserting plan files into WAE Live. This chapter describes both methods.

- Prerequisites
- Pre-Snapshot Configuration
- Using External Archives
- Inserting Plan Files Directly

This chapter references the following terms.

- `$CARIDEN_ROOT`—Location of the installation. By default, this is `/opt/cariden`.
- `$CARIDEN_HOME`—Directory in which the WAE Design, WAE Live, and WAE Collector executables and binaries are installed. The default is `/opt/cariden/software/mate/current`.

Note All instructions and examples assume you used `/opt/cariden` as the default installation directory. If you did not, then substitute your installation directory for `/opt/cariden`.

Prerequisites

This chapter does not describe the details of configuring snapshots. Rather, **it describes only the nuances of configuring manual snapshots for the purpose of discovering multiple networks**. Using this chapter has several "knowledge" prerequisites, as follows.

- How to configure both snapshot .txt and .inc files, and their relationships.
- How to configure manual snapshots, including steps not covered here, such as configuring snapshots to discover and model the network.
- Plan file insertion tools (`archive_insert` and `ml_insert_plan`). For information, refer to their `-help` output.

For information on configuring snapshot files and configuring manual snapshots, see Snapshot Files.

Best practice: Back up all configuration files before you begin.

Pre-Snapshot Configuration

Note This chapter uses two running examples. One is the collection for an "east" network where plan files are put into an external archive. The other is the collection for a "north" network where plan files are directly inserted in WAE Live. Such names are for example purposes only.

Note You can use the same authentication and network access files for all networks, or you can create and modify them on a per-network basis. For more information on these files, see Network Access File and Network Authentication.

Step 1 Run `mate_auth_init` once to create an authentication file (`auth.enc`) used by SNMP and login tools.

```
mate_auth_init
```

This is an interactive tool that first prompts you to choose the SNMP version and the relevant parameters. To create a different network authentication file for a network, use the `-auth-file` option. The recommendation is to use one of the default configuration paths: `~/.cariden/etc`, `$CARIDEN_HOME/etc`, or `$CARIDEN_ROOT/etc`.

Step 2 Optional: Customize network access. To create a different network access file, copy the default `$CARIDEN_HOME/etc/net_access.txt`, rename, and modify it. This file must be located in one of the default configuration paths: `~/.cariden/etc`, `$CARIDEN_HOME/etc`, or `$CARIDEN_ROOT/etc`.

Step 3 For new installations, copy the default snapshot.txt and snapshot.inc files to working configuration files. Uniquely name each set of .txt and .inc files to represent the network to which it is applicable.

Examples:

```
cp /opt/cariden/software/mate/current/etc/snapshot.txt /opt/cariden/etc/ss-east.txt
cp /opt/cariden/software/mate/current/etc/snapshot.inc /opt/cariden/etc/ss-east.inc
cp /opt/cariden/software/mate/current/etc/snapshot.txt /opt/cariden/etc/ss-north.txt
cp /opt/cariden/software/mate/current/etc/snapshot.inc /opt/cariden/etc/ss-north.inc
```

If this is not a new installation, you can use existing snapshot files in `/opt/cariden/etc` and make modifications noted in this chapter as needed. However, you need one set of snapshot files (one .txt file and one .inc file) per network.

Using External Archives

Each network must have its own set of snapshot files that independently call `archive_insert` to insert plan files into a uniquely named archive. Remember, you must also configure the snapshots to discover, poll, and build the network model.

Step 1 Edit the ss-east.txt file, which contains the collection, polling, modeling, and insertion tasks to perform. This file controls the sequence of execution and also contains environment variables of common values used in the ss-east.inc file.

a. At minimum, you must define `unique`, `seed_router`, `igp`, `home_dir`, and `archive_dir`. By default, the `archive_insert` tool uses the `archive_dir` environment variables when inserting plan files into an external archive. Best practice is to use the default.

Example:

```
unique east
seed_router 10.10.10.11
igp ospf
ospf_area 0.0.0
home_dir /opt/cariden
work_dir $(home_dir)/work
archive_dir $(home_dir)/archives
```

b. Edit the `include` environment variable to read the ss-east.inc file from `$(home_dir)/etc`.

Example: `include $(home_dir)/etc/ss-east.inc`

c. Uncomment the ARCHIVE_INSERT task.

Step 2 As needed, edit the ss-east.inc file to modify and add tools that are to be called from the ss-east.txt file.

Configure the file to use `archive_insert` to insert plan files into the named external archive directory. You can use the default `archive_insert` configuration in the .inc file for this purpose.

Step 3 Run `archive_init` to initialize the archive repository into which the plan files are to be inserted. Text in <angle brackets> refers to environment variables that you set in the ss-east.txt file. The path entered for the External Archive on the WAE Live Settings > General Settings page must match this case-sensitive path.

```
archive_init -archive /opt/cariden/archives/<unique>-archive
```

Example: `archive_init -archive /opt/cariden/archives/east-archive`

Step 4 Create a cron job that repeats the process of running the snapshot files that you created, which results in the insertion of plan files going into their respective archive repositories.

Note Both CARIDEN_ROOT and CARIDEN_HOME variables must be defined from within the crontab. You cannot use CARIDEN_HOME=$CARIDEN_ROOT/software/mate/current.

a. Open the file for editing as follows.

```
crontab -e
```

b. At the end of the file, add the following lines.

```
CARIDEN_ROOT=/opt/cariden
CARIDEN_HOME=/opt/cariden/software/mate/current
```

c. At the end of the file, add one entry to call the unique snapshot per network.

Example:

```
*/30 * * * * $CARIDEN_HOME/bin/snapshot -config-file $CARIDEN_ROOT/etc/ss-east.txt
2>&1
```

Note Best practice is to verify the snapshots are working prior to continuing to WAE Live configurations.

Inserting Plan Files Directly

Each network must have its own set of snapshot files that independently call `ml_insert_plan` to insert data directly into the WAE Live data store. These files must also call `archive_insert` to insert plan files directly into the Map archive. Remember, you must configure the snapshots to discover, poll, and build the network model.

Step 1 Edit the ss-north.txt file, which contains the collection, polling, modeling, and insertion tasks to perform. This file controls the sequence of execution and also contains environment variables of common values used in the ss-north.inc file.

a. At minimum, you must define `unique`, `seed_router`, `igp`, `and home_dir`.

If using the Map component, create an environment variable that uniquely specifies the Map archive.

Example:

```
unique north
seed_router 10.10.10.11
igp isis
isis_level 2
home_dir /opt/cariden
```

```
work_dir $(home_dir)/work
map_archive_dir $(home_dir)/data/mldata/archive
```

b. Edit the `include` environment variable to read the ss-north.inc file from `$(home_dir)/etc`.

Example: `include $(home_dir)/etc/ss-north.inc`

c. Uncomment the ML_INSERT task.

d. If using the Map component, add an MAP_ARCHIVE_INSERT task.

Example:

```
<ARCHIVE_INSERT_TASKS>
#ARCHIVE_INSERT
ML_INSERT
MAP_ARCHIVE_INSERT
```

Step 2 As needed, edit the ss-north.inc file to modify and add tools that are to be called from the ss-north.txt file.

Configure the associated ss-north.inc file to use `ml_insert_plan` with the `-network` option. The `-network` option specifies the network partition of the data store into which you are placing data. This must match the case-sensitive network name added through the WAE Live UI. In this example, you would also have to create and name a WAE Live network called “north.”

Example:

<ML_INSERT>	
Name	Value
cmd	$(cariden_home)/bin/ml_insert_plan
cmd_opt	ML_INSERT_CMD_OPT
<ML_INSERT_CMD_OPT>	
Name	Value
plan-file	$(work_dir)/$(unique).pln
time	$(start_time_direct)
log-file	$(log_dir)/$(unique)-log-ml_insert.log
network	north

Step 3 If using the Map component, configure the associated ss-north.inc file to use `archive_insert` to specify the Map archive. This must match the case-sensitive name given the Map archive in the WAE Live Settings > General Settings page. In this example, the Map archive in WAE Live would have to be `/opt/cariden/data/mldata/archive/north`.

Example:

<MAP_ARCHIVE_INSERT>	
Name	Value
cmd	$(cariden_home)/bin/archive_insert
cmd_opt	MAP_ARCHIVE_INSERT_CMD_OPT

<MAP_ARCHIVE_INSERT>	
<MAP_ARCHIVE_INSERT_CMD_OPT>	
Name	Value
plan-file	$(work_dir)/$(unique).pln
archive	$(map_archive_dir)/$(unique)
time	$(start_time)
log-file	$(log_dir)/$(unique)-log-map-archive_insert.log

Step 4 Create a cron job that repeats the process of running the snapshot files that you created, which results in the insertion of plan files going into their respective network segments within the WAE Live data store.

Note Both CARIDEN_ROOT and CARIDEN_HOME variables must be defined from within the crontab. You cannot use CARIDEN_HOME=$CARIDEN_ROOT/software/mate/current.

a. Open the file for editing as follows.

```
crontab -e
```

b. Add the following lines.

```
CARIDEN_ROOT=/opt/cariden
CARIDEN_HOME=/opt/cariden/software/mate/current
```

c. At the end of the file, add one entry to call the unique snapshot per network.

Example:

```
*/30 * * * * $CARIDEN_HOME/bin/snapshot -config-file $CARIDEN_ROOT/etc/ss-north.txt
2>&1
```

Best practice is to verify the snapshots are working prior to continuing to WAE Live configurations.

Offline Discovery

Note Configuring `get_configs` is supported in both augmentation and manual collection methods. Using `parse_configs` to augment a plan with RSVP LSP and/or SRLG data is supported through the augmentation method. Configuring `parse_configs` to create a plan file for overall topology is supported only in manual collection.

This chapter describes the CLI tools available to discover and retrieve information from router configuration tools and from RRD tools, such as Cricket, Cacti, and MRTG.

Import Databases

The following tools are useful for capturing and importing network information. For instance, you can capture the configuration files or IGP databases and import them into WAE Collector.

- `get_configs`—Reads the configuration files from a list of routers and saves them in the specified directory.
- `parse_configs`—Reads a set of Cisco and/or Juniper Networks router configuration files, and creates a plan file of the network. See Import Router Configuration Files. For information on using this tool from the WAE Design GUI, see the *Cisco WAE Design Integration and Development Guide*.
- `parse_igp`—Converts IGP information from router `show` commands to a plan file. See Import IGP Database. For information on using this tool from the WAE Design GUI, see the *Cisco WAE Design Integration and Development Guide*.
- `get_show` and `get_xml`—Tools for entering router `show` commands and the XML equivalents for further processing by the user or an application. These commands are typically used because the WAE Collector does not include the output of the commands during plan file creation. The `get_xml` tool offers similar functions to `get_show`. It is used to get structured data from devices by executing XML commands on them. The command format is device-dependent.

Note This section contains examples for Cisco and Juniper routers. For information about network discovery of routers for other vendors, please contact your support representative.

Import Router Configuration Files

The `parse_configs` tool reads Cisco, Juniper, and Huawei router configuration files, and creates a plan file.

The router configuration files from the network, or part of the network, need to be available in a specific directory. The `parse_configs` tool reads files in this directory (`-data-dir` option), determines the router type/vendor, and parses the configuration.

The following information can be read from a router configuration file to create the plan file. After parsing this information, the tool matches corresponding interfaces in the IGP mesh to create the network topology.

• Router name • Vendor • Model • OS • Router IP address (loopback) • Management interface IP address (Cisco IOS XR and Juniper Junos) • Interface names (inside IGP topology) • Interface IP addresses • Interface capacities (if available)	• IGP type and metrics (IS-IS or OSPF) – Process ID (OSPF) – Instance ID (IS-IS) • RSVP reservable bandwidth (MPLS) • MPLS LSPs (including bandwidth and FRR LSPs) • LAG[1] ports and bundle ports • SRLGs, including which SRLGs are configured on which nodes (Cisco and Juniper) • VPN interfaces • VPN PE membership

1. LAG is specific to Ethernet. Bundle is generic and applies to different link types.

With the `-igp-protocol` option, you can select which interfaces are part of the topology: IS-IS and/or OSPF enabled interfaces. The default is `isis`.

- For IS-IS networks, the tool can read IS-IS Level 1, Level 2, or both Level 1 and Level 2 metrics. If both are selected, `parse_configs` combines both levels into a single network, and Level 2 metrics take precedence. The `-isis-level` option specifies which option to use; the default is Level 2.
- For OSPF networks, the tool can read information for single or multiple areas. The `-ospf-area` option specifies the area ID or all. The default is area 0.

ASN is ignored by default. However, for networks that span multiple BGP ASNs, use the `-asn` option to read information from more than one IGP process ID or instance ID in an ASN.

Shared media segments in the network (non point-to-point circuits, such as Ethernet) are included in the topology by default unless the `-shared-media` option is set to `false`. A pseudonode and interface representing the medium are then created for every shared medium with more than two hosts, as used by OSPF and IS-IS routing protocols.

With the `-plan-file` option, you can merge an existing plan file with router configurations to create an augmented plan file. For example, you could use the `parse_igp` output as the input into `parse_configs`.

Note A useful tool for maintaining an archive of router configuration files is RANCID (`http://www.shrubbery.net/rancid/`).

Import IGP Database

The parse_igp tool reads one or more databases that are generated from a router's CLI. With the `-igp-protocol` option, you can select an IGP protocol.

- IS-IS IPv4 or IPv6 using the `isis` or `isisv6` option, respectively
- OSPF IPv4 or IPv6 using the `ospf` or `ospfv3` option, respectively

With the `-plan-file` option, you can merge an existing plan file with the IGP databases to create an augmented plan file. For example, you could use the `parse_configs` output as the input into `parse_igp`.

IS-IS

This tool can generate a topology out of an IS-IS Level 1, Level 2, or both databases using the `-level` option.

To capture an IS-IS database from the CLI, log into a router within the IS-IS topology, display the IS-IS database, and save the output of that session to a file, as follows.

Step 1 Establish a terminal session on a host that has direct access to the network routers, for example, using telnet or SSH.

Step 2 Initiate a process to capture the entire session.

Step 3 Log on to the seed router, which is a router that contains IGP information for the network.

Step 4 Disable paging of output by setting the terminal length to infinite (0).

Cisco:	`terminal length 0`
Juniper:	`set cli screen-length 0`

Step 5 Cisco Option: Disable dynamic host resolution in IS-IS if hostnames are longer than 14 characters and the unique part of the name is after 14 characters. Cisco routers truncate names at 14 characters. To disable dynamic host resolution, enter the `router isis <proc id>` command mode, and then enter:

```
no hostname dynamic
```

Step 6 Display the IS-IS link-state database (LSDB).

Cisco:	`show isis database verbose`
Juniper:	`show isis database extensive`

Step 7 Log out of the router, the network host, and the screen capture, each time using the `exit` command.

Step 8 Save your session capture that was initiated in step #2 (`exit` when using `script`).

Repeat the above steps to capture IS-IS databases from additional routers (Level 1 and Level 2), if necessary. The resulting file or directory includes login and logout commands, as well as output. Now you can use the `parse_igp` tool.

- Use the `-level` option to specify whether discovering Level 1 or Level 2 topology or both; the default is level 2.
- Pass the file created in the above steps using the `-database-file` option.
- If there are multiple files (multi-level topologies), pass the directory name where the files are located using the `-database-dir` option.

Example: This command uses IS-IS Level 2 topology information stored in the `mobile_database.txt` file to create a plan file called `mobile_model.txt`.

```
parse_igp -igp-protocol isis -database-file mobile_database.txt -out-file mobile_model.txt
```

IS-IS Database Information

By default, the IS-IS protocol used in IP networks (non MPLS) does not distribute the IP addresses of the interfaces in the network, nor the circuit capacities.

When the IS-IS TE-extensions (for MPLS) have been enabled in the network, that information becomes available, and will also be used by `parse_igp`.

- Cisco—You must specifically enable the TE extensions using `mpls traffic-eng level-2` in the `router isis` configuration section, and `mpls traffic-eng tunnels` on the interfaces. Doing so makes both the IP addresses and circuit capacities available in IS-IS (and `parse_igp`).
- Juniper Networks—IS-IS TE extensions are enabled by default, and IP addresses are available for all interfaces in those routers. If RSVP is also enabled on an interface, the capacity of that circuit is available in IS-IS.

Enable the `-use-dns` option by setting it to `true` if DNS (domain name server) needs to resolve IP addresses (router names) in the IS-IS database file.

Note Parallel circuits (non-TE enabled) between two Cisco routers, show up in the IS-IS database as a single circuit.

OSPF

To capture an OSPF database from the CLI, log into a router with the OSPF topology, display the OSPF database, and save the output of that session to a file.

Step 1 Establish a terminal session on a host that has direct access to the network routers, for example, using telnet or SSH.

Step 2 Initiate a process to capture the entire session.

Step 3 Log on to the seed router. This is a router that contains IGP information for the network.

Step 4 Disable paging of output by setting the terminal length to infinite (0).

Cisco: `terminal length 0`

Juniper Networks: `set cli screen-length 0`

Step 5 Follow the appropriate Cisco or Juniper Networks step.

Cisco	Juniper Networks
If the system supports DNS, enable it so the database includes host names, rather than just IP addresses. Enter the configuration command mode, and then enter this command. IOS: `ip ospf name-lookup` IOS XR: `ospf name-lookup`	Display the OSPF database. If the system supports DNS, pipe the output of the `show` command to the resolver so the database has host names, rather than just IP addresses. `show ospf database extensive | resolve`
Display the OSPF database. IOS: `show ip ospf database router` IOS XR: `show ospf database router`	Otherwise, just show the database, which identifies routers by IP address only. `show ospf database extensive`
Display the OSPF TE database. IOS: `show ip ospf database opaque-area` IOS XR: `show ospf database opaque-area`	Display OSPFv3 database. `show ospf3 database extensive`

Display OSPFv3 database (IOS only)
`show ipv6 ospf database router`
`show ipv6 ospf database link`
`show ipv6 ospf database prefix`	

Step 6 Log out of the router, the network host, and the screen capture, each time using the `exit` command.

Step 7 Save your session capture that was initiated in step #2 (`exit` when using `script`).

Repeat the above steps to capture OSPF databases from additional area border routers (ABRs), if necessary. The resulting file or directory includes login and logout commands, as well as output. Now you can pass the file created in the above steps to using the `parse_igp` tool using the `-database-file` option. If there are multiple files, then pass the directory name where the files are located using the `-database-dir` option.

By default, `parse_igp` collects the OSPF area 0 link-state database (LSDB). To generate topologies from non-zero area LSDBs, use the `-ospf-area all` option. The tool then identifies all ABRs and builds a complete multi-area OSPF network topology. Note that the `login_find_igp_db` tool uses this `-ospf-area all` option as well.

OSPF Database Information

Unlike the IS-IS database, the OSPF database has IP address information for all interfaces in the network. If the network is TE-enabled, the OSPF database also contains circuit capacities.

Enable the `-use-dns` option by setting it to `true` if DNS needs to resolve IP addresses (router names) in the IS-IS database file.

Note Parsing IGP with the OSPF protocol option only processes area 0 routers per default. Use the `-area` option to select another area, or `all` for all areas.

get_show and get_xml

The `get_show` tool is a wrapper for entering a `show` command on one or more routers. For example, the `get_show` tool with a `-cmd` argument of `show configuration` is equivalent to the `get_configs` tool. The `-command-table` option enables you to enter vendor-specific CLI commands, such as an ICMP ping in multi-vendor networks. You could also use this tool to get an OSPF or IS-IS database from the router.

Because `show` commands are highly dependent on router types, this tool can only operate on a homogeneous set of routers when more than one is specified. The IS-IS and OSPF `show` commands are listed in the IS-IS and OSPF sections, respectively.

In the `-nodes-table` or `-nodes` arguments, if an IP address is available, it is used. Otherwise, an IP lookup through DNS is tried. If that fails, an error is returned.

The `get_xml` tool offers the same function as `get_show`. It is used to get structured data from devices by executing XML commands on them. The command format is device-dependent.

Import Traffic from RRD Tools

WAE Collector can import network information from the following RRD tools.

- Cricket
- Cacti
- MRTG

Cricket

The `cricket_poll_interfaces` tool reads a router interfaces file, discovers which interfaces the file specifies, and the RRD files that contain the data associated with each interface. It then reads the traffic measurements from the RRD file and imports them into to the <InterfaceTraffic> and <NetIntIfMeasurements> tables in a plan file.

Cacti

Because Cacti is written in PHP and uses mysql, the importer is also implemented with a PHP script. Install the PHP script on the web server that is running Cacti, and then invoke `cacti_poll_interfaces` to import the traffic measurements into the <InterfaceTraffic> and <NetIntIfMeasurements> tables in a plan file.

```
http://.../cacti/graph_info.php?get=tab
```

To install the PHP script on a web server running Cacti, follow these steps.

Step 1 Copy the PHP script to the web server. You must have a guest account set up for Cacti. The script location is as follows.

```
$CARIDEN_HOME/lib/php/cacti/graph_info.php
```

Step 2 Add `"graph_info.php" => 7`, to `include/global_arrays.php`. The array location is as follows.

```
$user_auth_realm_filenames
```

To import traffic measurements into a plan file, call `cacti_poll_interfaces` and provide the Cacti URL as an argument.

```
http://.../cacti/graph_info.php?get=tab
```

MRTG

The `mrtg_poll_interfaces` tool imports traffic measurements into a plan file by reading an MRTG configuration file. First it discovers which interfaces the configuration file specifies, along with the RRD files that contain the data associated with each interface. Then it reads the RRD files and imports the traffic measurements into the <InterfaceTraffic> and <NetIntIfMeasurements> tables in a plan file.

Network Access File

Note Editing the network access file is supported for the WAE Collector UI collection and for the manual collection methods.

A network access file can be used to store network access parameters. These include timeout and retry settings, and settings for management of multiple simultaneous queries. Having these settings in a file, rather than as CLI parameters, removes the redundancy across many calls and allows for more complex settings (per router settings, for example).

The network access file provides default settings for all access parameters. You can use either the default network access file, or you can modify and put it in one of the following locations. The file is looked for in this sequence, and the first version found is used.

- `~/.cariden/etc`
- `$CARIDEN_ROOT/etc`
- `$CARIDEN_HOME/etc` (default)

When the Collector server uses this file, it saves it as net_access_session.txt file. Augmented snapshots then use the net_access_session.txt file that was us chapter.

Best Practices

- Make a copy of the default net_access.txt file located in `$CARIDEN_HOME/etc` before editing it, and ed in the last collection by the Collector server. For information on configuring the Collector server, see Collecting Basic Information Using the WAE Collector UI and put the edited version in `$CARIDEN_ROOT/etc`. This simplifies the upgrade process and preserves a copy of the original if needed.
- When upgrading, compare the net_access.txt file in the new release to the one in the existing release to determine if your edits need to be incorporated into the new net_access.txt file.

File Format

The network access file consists of two sections: one containing tables that set values globally and one containing tables that sets values on a per-router basis.

Note In the net_access.txt file an empty field means *everything else*, and this meaning is in context of the rows defined before it. If it is in the first row, it means *everything*.

Global Settings

WAE Collector network communication tools take advantage of the polling abilities that simultaneously process a large number of network requests. The Global section of the network access file defines constraints that are used to limit the impact to either the server doing the polling or to the network elements between the server and the network being polled. Examples of network elements that could be heavily impacted by polling traffic are a firewall, slow WAN circuits, or a NAT device.

This section consists of two tables that work in tandem: <GlobalModes> and <GlobalSettings>.

- <GlobalModes>—This table groups together settings that are used to constrain the speed of the network communications. These settings are grouped into names (in the Name column), and are activated by referencing them in the GlobalMode column of the <GlobalSettings> table. These names are user-definable.

  The network access file includes commented documentation for each <GlobalModes> property. Table 4-1 provides an example <GlobalModes> table.

- <GlobalSettings>—This table defines the association between the entries in its TaskRegExp column and the entries in the <GlobalModes> Name column.
  - TaskRegExp—This is the WAE Collector CLI tool. The default is a blank, which matches all possible tools.
  - GlobalMode—Mode to assign to all routers when running the matched CLI tool.

Table 4-2 provides an example <GlobalSettings> table. The empty field at the beginning of the last row means *everything except* `snmp_poll` and `snmp_find_*`.

Table 4-1 Example <GlobalModes> Entries

Name	Property	Value
Normal	SNMP_max_queries_total	1000
Normal	SNMP_max_open_session	200
Normal	SNMP_collection_interval	120000
Normal	LOGIN_max_open_sessions	10
Normal	LOGIN_session_open_interval	0
Slow	SNMP_max_queries_total	500
Slow	SNMP_max_open_session	50
Slow	SNMP_collection_interval	240000
Slow	LOGIN_max_open_sessions	2
Slow	LOGIN_session_open_interval	1
Fast	SNMP_max_queries_total	2000
Fast	SNMP_max_open_session	400
Fast	SNMP_collection_interval	60000
Fast	LOGIN_max_open_sessions	20
Fast	LOGIN_session_open_interval	0

Table 4-2 Example <GlobalSettings> Entries

TaskRegExp	GlobalMode
snmp_poll	Fast
snmp_find_*	Slow
	Normal

Per Router Settings

If you have concerns about specific device types or operating systems, you can constrain the WAE Collector network communication tool to execute on a per-router basis. For example, some devices might not respond well to short SNMP timeout values when they are busy, while others might need special settings for login access. Together, the <RouterModes> and <PerRouterSettings> tables enable you to adjust these types of settings.

- <RouterModes>—This table defines groups of devices to either block or constrain their communications. For each name (in the Name column) that you create, you must enter a value for all SNMP properties.

  The network access file includes commented documentation for each <RouterModes> property. Table 4-3 provides an example <RouterModes> table.

- <PerRouterSettings>—This table associates named groups of <RouterModes> parameters with a specific set of devices within the network. Each Name entry in the <RouterModes> table has a corresponding entry in the RouterMode column.

  Each RouterMode is defined by the NodeRegExp, IPRegExp, and SQLFilter columns.

  - NodeRegExp is matched against device names.
  - IPRegExp is matched against device IP addresses.
  - SQLFilter is an SQLite `sql` command that can reference any column of the Nodes table to match devices.

  The TaskRegExp column provides constraints for one specific tool in the event that unique parameters are required for one discovery task.

  Table 4-4 provides an example <PerRouterSettings> table. The empty fields in the first row mean *everything*. The empty TaskRegExp field in the last row means *everything except* `snmp_find_multicast` and `snmp_poll`.

Table 4-3 Example <RouterModes> Entries

Name	Property	Value
Normal	SNMP_max_timeout	3
Ignore	SNMP_max_timeout	0
Limit_CRS	SNMP_max_timeout	3
Multicast_Login	SNMP_max_timeout	3
Multicast_SNMP	SNMP_max_timeout	3
Junos_old	SNMP_max_timeout	3
Junos_new	SNMP_max_timeout	3
Normal	SNMP_RSVP_stats_method	Default
Junos_new	SNMP_RSVP_stats_method	Method1
Junos_old	SNMP_RSVP_stats_method	Method2

Table 4-4 Example <PerRouterSettings> Entries

NodeRegExp	IPRegExp	TaskRegExp	RouterMode	SQLFilter
			Ignore	Name REGEXP '^sl-gw.*'
		snmp_find_multicast	Ignore	Name NOT REGEXP 'sl-crs.*' AND Name NOT REGEXP 'sl-bb.*'
		snmp_poll	Limit_CRS	OS REGEXP '^IOS XR.*'
			Normal	

Discovering Multi-Vendor Networks with SAM

If you are discovering a network containing both Alcatel and non-Alcatel nodes, you must configure the <PerRouterSettings> table to tell the online tools to ignore the Alcatel objects and their traffic. The simplest method is to do the following.

Step 1 Add a comment (#) to this line to prevent the collection of Alcatel statistics.

```
# snmp_pollALU_REALTIMEOS REGEXP '^TiMOS.*'
```

Step 2 Uncomment this line to ignore the discovery of Alcatel nodes, interfaces, and LSPs, and to ignore the collection of statistics from them.

```
(snmp_find_nodes|snmp_find_interfaces|snmp_find_rsvp|snmp_poll)IgnoreVendor =
'Alcatel-Lucent'
```

Test the Network Access File

The `mate_access_test` tool enables you to specify a node, node IP, and task, or alternatively specify the router mode and global mode settings directly. The tool returns the global and per-router parameter settings that are applied if the network access file were used. The option is `-net-access-file`. The default value is net_access.txt.

Use `mate_access_test -net-access-file` to see the global and per-router parameter settings that are applied if a network access file is specified. The default value for `-net-access-file` option refers to the net_access.txt file in the configuration path.

Tool Access Parameters

Each WAE Collector online tool (for example, `snmp_find_interfaces`) contains three parameters to control network access settings.

- `-net_access_file <file>`—Overrides the default network access file.
- `-net_access-router-mode <name>`—This name specifies the RouterMode that overrides the <PerRouterSettings> table.
- `-net-access-global-mode <name>`—This name specifies the GlobalMode that overrides the <GlobalSettings> table.

Network Authentication

Note Creating and editing the authentication file is supported for the manual collection method.

The authentication file consolidates the login, authentication, encryption, community strings, and other credentials needed by the WAE Collector tools to access routers and collect network data. It is required if the tools are to be called by scripts, or if different routers in the network require different authentication information. The file can be encrypted for security and protected with a master password.

- Manual snapshots—Use the auth.enc authentication file. The `mate_auth_init` tool simplifies the process of creating a default authentication file.
- Augmented snapshots—Use the auth_session.enc file that was used in the last collection by the Collector server. The password for de-encrypting this file is set in the Node Access page of the WAE Collector UI.

Online Discovery Authentication

When an online discovery tool needs authentication information for a router (for example, `snmp_find_interfaces` needs a community string to perform an SNMPv2c query), it accesses the authentication file and looks for a match for the router. If successful, the tool uses the credentials from the file to access routers and collect network data. Without a match the tool generates a prompt or notification.

- SNMPv2c—Prompts for authentication credentials and proceeds.
- SNMPv3—Notifies the user to create an authentication file and terminates.

You can disable user interaction by setting the `-auth-prompt` option to `false`.

Create an Authentication File

Note Use this method of creating a network authentication file only if using the manual snapshot collection process.

The `mate_auth_init` tool is an interactive tool that simplifies the process of specifying a default set of authentication credentials that WAE Collector tools use to access all routers. The file is created in the directory from which you execute the command. To change the file location, enter a full path name.

The file it creates has credentials for SNMPv2c, SNMPv3, or both. SNMPv2c uses a less secure security model, passing community strings in clear text. SNMPv3 provides a strong security model that supports authentication, integrity, and confidentiality.

If `mate_auth_init` does not find an auth.enc file in one of the default locations, the tool prompts you to select one from a list.

- `~/.cariden/etc`
- `$CARIDEN_HOME/etc`
- `$CARIDEN_ROOT/etc` (Linux only)

The tool creates a file named `auth.enc` in the selected directory. However, you can override the default directory and filename by using the `-auth-file` option. The recommendation is to use one of the above default configuration paths. If you put this file in a different directory, binaries must be explicitly called using this path.

Example: `mate_auth_init -auth-file /opt/cariden/etc/auth-acme.enc`

The `mate_auth_init` tool prompts you to choose SNMPv2c, SNMPv3, or both. Depending on your choice, the tool prompts you for authentication information that is pertinent to the selected SNMP version.

Note If both SNMPv2c and SNMPv3 are selected, the default is for the auth.enc file to put all nodes in both SNMPv2c and SNMPv3. When a node is mapped to both, then only SNMPv3 is used. To change this behavior, decrypt the auth.enc file using `mate_auth_export`, edit the authentication tables based on the IPRegExp values, and then re-import the file using `mate_auth_import`.

The authorization file password and default seed router login credentials consist of the following.

- master password—Password for viewing file contents
- login username—Default username for login access to the routers

- login password—Default password for login access to the routers
- login enable password—Default enable password for login access

The SNMPv2c information is defined using a single value.

- community—Default community string

The SNMPv3 information defines authentication and encryption details.

- Security level
  - noAuthNoPriv—Authenticates by username, but does not validate the user and does not encrypt data.
  - authNoPriv—Authenticates by username, validates the user using MD5 or SHA, but does not encrypt data.
  - authPriv—Authenticates by username, validates the user using MD5 or SHA, and encrypts data using DES or AES.
- SNMPv3 username—Username for authentication
- Authentication protocol type—MD5 or SHA
- Authentication password—Password for authentication
- Encryption protocol type—DES or AES
- Encryption password—Password for encryption
- Context name—Name of a collection of management information accessible by an SNMP entity

After you have created the initial encrypted authentication file, you can manually edit the contents to add multiple profiles or communities and map routers to them. Each profile contains a complete set of SNMPv3 authentication and encryption information. Multiple profiles or communities are necessary when different groups of routers use different authentication credentials. For information about editing an encrypted authentication file, see Add Router-Specific Authentication Information.

Tables in the Authentication File

The contents of the encryption file are organized into tables.

- <MasterPassword>—Contains the master password for viewing or changing the file (Table 4-5).
- <UserTable>—Contains usernames and passwords for login access to nodes (Table 4-6).
- <CommunityTable>—Contains SNMPv2c community strings for access to nodes (Table 4-7).
- <SNMPv3ProfileTable>—Contains SNMPv3 profiles, which define a set of authentication, encryption, and context information (Table 4-8).
- <SNMPv3MappingTable>—Defines how to match routers with the SNMPv3 profiles (Table 4-9).

Note If both SNMPv2c and SNMPv3 are selected, the default is for the auth.enc file to put all nodes in both SNMPv2c and SNMPv3. When a node is mapped to both, then only SNMPv3 is used. To change this behavior, decrypt the auth.enc file using `mate_auth_export`, edit the authentication tables based on the IPRegExp values, and then re-import the file using `mate_auth_import`.

Table 4-5 <MasterPassword> Format

Column	Description
Password	Optional: Password for accessing the authentication file. If table or password is missing, the authentication file is unencrypted plain text and no password is required to view or change the file contents.

Table 4-6 <UserTable> Format

Column	Description
IPRegExp	Regular expression to match node IP addresses; if missing, defaults to accept all.
Username	Username for login access.
Password	Password for login access.
EnablePassword	Enable password for login access.

Table 4-7 <CommunityTable> Format

Column	Description
IPRegExp	Regular expression to match node IP addresses; if missing, defaults to accept all.
Community	Community string for SNMP access.

Table 4-8 <SNMPv3ProfileTable> Format

Column	Description
Profile Name	Descriptive name of the routers to which this profile applies.
Security Level	Level of SNMP security. Value is noAuthNoPriv, authNoPriv, or authPriv.
Username	User for which SNMP services are provided.
Auth Protocol	Protocol for authenticating the user. Values are MD5 or SHA. Required if using authNoPriv or authPriv security level.
Auth Password	Authentication password. Required if using authNoPriv or authPriv security level, and must be equal to or greater than eight characters.
Encryption Protocol	Protocol for encrypting data. Values are DES or AES. Required is using authPriv security level.
Encryption Password	Encryption password. Required if using authPriv security level, and must be equal to or greater than eight characters.
Context Name	Optional: A collection of management information accessible by an SNMP entity. If one or more context names are configured on a router, then a value is required. You can enter one context name only, and it is used to access all routers.

Table 4-9 <SNMPv3MappingTable> Format

Column	Description
IPRegExp	Regular expression to match node IP addresses; if missing, defaults to accept all.
Profile Name	Name of a profile in the <SNMPv3ProfileTable>.

Add Router-Specific Authentication Information

You can add additional router-specific information to the authentication file by adding rows to the authentication file tables (see Tables in the Authentication File). For router login and authentication, edit the <UserTable>. For SNMP, the following tables apply.

- SNMPv2c—Add community strings to the <CommunityTable> and map routers to these communities with a regular expression in the `IPRegExp` column.
- SNMPv3—Add security profiles to the <SNMPv3ProfileTable> and map routers to profiles in the <SNMPv3MappingTable> with a regular expression in the `IPRegExp` column.

If the authentication file is encrypted using a master password, you must first export the contents to plain text using the `mate_auth_export` tool, edit the tables using a text editor, and then encrypt it using `mate_auth_import`.

For SNMPv2c communities only, a more convenient method is provided by `auth_try_communities`. First provide a list of nodes (routers), for example from a plan file obtained through parsing the IGP database. You are then prompted for a number of communities to try. The tool attempts SNMP access to all the routers using each of the communities. If any routers are accessed successfully, these communities are entered in the authentication file to match the router names.

You can run the `auth_try_communities` tool repeatedly to add communities to the authentication file.

Note There is no equivalent tool for SNMPv3.

View Authentication Information

You can view the entire contents of the authentication file using the `mate_auth_export` tool, which exports a decrypted version of an authentication file. You can also view authentication information for a specific router using the `mate_auth_test` tool. Either way, you need the master password to view the contents.

Test the Authentication File

Test the authentication file using one of these tools.

- `mate_auth_test`—Prints authentication credentials for a specified authentication file, for a specified node IP address. The output returns whether the lookup is successful or optionally, shows all authentication details in plain text.
- `snmp_test`—Tests access to a specified router by sending a ping and an SNMP query using the credentials in the authentication file. If both SNMPv2c and SNMPv3 are present, then SNMPv3 is used.

- `login_test`—Tests login access to a specified router; in doing so it tests the login information provided by the authentication file.

Manage Archives

Note The tasks of configuring a plan file archive repository and inserting plan files into the archive are supported in both augmentation and manual collection methods. If you are collecting data only by configuring the WAE Collector web UI, then the archives described in this chapter are not applicable.

An *archive* is a repository containing network plan files, specific data items for plotting, and other data that is collected through augmented or manual snapshots. Additionally, information can be added to archives using CLI tools outside the snapshot process.

This chapter describes the basic archive tools.

Create or Update an Archive

Use `archive_init` to either create a new archive repository or to update the file structure of an existing archive.

- To create a new archive, set the `archive_init -archive` parameter to the path and name of the directory that will hold the archive. This creates a new, empty archive. The structure and support files for the archive are not complete until after the first recorded insertion.
- To update the file structure of an existing archive to that of the latest release, set the `archive_init -archive` parameter to the directory of an existing archive and set `-upgrade` option to `true`. This updates the file structure of the existing archive to that of the latest release.
- **For manual configurations of WAE Live**, you must override the default data typically extracted for an archive in order to create the Events panel.

```
archive_init -archive <Map Archive Path> -timeplot-summary-format
$CARIDEN_HOME/.cariden/etc/matelive/default_timeplot_summary_format.txt
```

Update Summary of Time-Sequence Plot Data

Use `archive_update` to update the summary of time-sequence plot data stored in an archive, for example after changing the summary format file.

- Set the `-archive` parameter to the location of an existing archive.
- Set the `-timeplot-summaries` parameter to `true`.
- Set the `-start-time` parameter to the timestamp of the first record to update.

By default, this tool updates all records from the start time stamp to the end of the archive, however, you can optionally specify the `-end-time`.

For information on configuring the time-sequence plot data, see the *Cisco WAE Design Archive User and Administration Guide*.

Insert or Extract Files from an Archive

Each archive record can contain one or more of the following files.

- Network plan file (.pln) obtained using the `snapshot` tool.
- Time-sequence plot summary file (.sum), automatically constructed using default summary format settings. This file can also be constructed and inserted manually. (See the *Cisco WAE Design Archive User and Administration Guide*.)
- Optional: User file or any other file.

For WAE Design Archive, the archive can also contain a visual format file, which specifies how the time-sequence plot data should be displayed in the web browser.

Insert Files

You can insert files all at once using one `archive_insert` tool, or individually using multiple CLI tools. All files in the archive repository are stored and accessed using a timestamp, so unless you want to use the default current timestamp when adding files to the archive, you include the `-time <timestamp>` option.

Note If the `snapshot` tool is configured to generate .txt format plan files, use the `mate_convert` tool before `archive_insert` to convert the .txt format plan file to the .pln format plan file.

Inserting a plan file into the archive automatically updates the files needed for interacting with the archive information via the web browser. For this reason, do not copy a file into the archive directory.

You can also insert WAE Design Archive plan files into the archive by choosing **File > Save to > Design Archive** in the WAE Design GUI. For information, see the *Cisco WAE Design Archive User and Administration Guide*.

Extract and Delete Files

After files have been archived, you can retrieve a copy using the `archive_extract` tool. CLI options specify which files to retrieve, where to copy them, and what to name them. You must include a timestamp. However, you can also specify that WAE Collector use the closest time to the timestamp provided, or you can specify a range of time to get a batch of files.

To retrieve user files with `archive_extract`, follow one of these options.

- Specify the name of the file to extract, or a partial name with wildcards (*), with the `-user-files` option.
- Specify a list of file names with the `-user-files-list` option.

You can also use `archive_extract` to remove items from the archive. The procedure is the same as for extracting files, except that you use the `-delete` parameter to delete the file after extraction. This process ensures that you always have a local copy of files that you delete, in case the deletion was accidental or incorrect.

You can also retrieve plan files from the archive by choosing **File > Open from > Design Archive** or **File > Open from > WAE Live** in the WAE Design GUI. For information, see the *Cisco WAE Network Visualization Guide*.

Manage Archives for WAE Design Archive

The `archive_config` tool enables you to manage the archive repositories available to the WAE Design Archive application on the web server. This tool creates an `archivelist.xml` file in the `$CARIDEN_HOME/etc/archive/config` directory.

- `-action add`—Add the archive repository to a specific location.
- `-name`—Name of the archive repository.
- `-path`—Full path of the archive repository.
- `-template-dir`—Full path of the template.
- `-template-name`—Name of the template used by all files in this archive repository.

Example: This example adds an archive named SW_Region that has a path of `acme/archives/acme_backbone`. The template directory is `acme/data` and the template name is `acme_backbone-template.pln`.

```
archive_config -action add -name SW_Region -path acme/archives/acme_backbone -template-dir
acme/data -template-name acme_backbone-template.pln
```

Note The `archive_config` CLI tool and the WAE Design GUI Archive feature do not apply to WAE Live.

Make Batch Changes to Archive Files

Maintenance of archives sometimes requires similar updates to multiple files in an archive. Here are two examples:

- A change in topology requires application of a new template file to the plan files between two time stamps
- A change in reporting requirements requires an updated summary file for plans for all plans in the archive.

You can perform this task with individual CLI tools, or in many cases you can use the `archive_do` tool to consolidate the CLI tools. The `archive_do` tool gets a list of timestamps between `-time` and `-time-to`, using `archive_extract`, and then performs the following for each timestamp.

- Uses `archive_extract` to extract all `%extract_*` files into a local directory.
- Executes CLI tools in `-cmd` argument sequence in the local directory. Table 4-10 lists the valid variables in the `-cmd` argument.
- Uses `archive_insert` to insert all `%insert_*` files into the archive.

The `archive_do` tool creates a list of CLI calls for all timestamps, fills in the temporary files at each step, and surrounds the calls with the relevant `archive_extract` and `archive_insert` tools. You can view the CLI tools without applying them to an archive by specifying the `-dry-run` option.

Table 4-10 Valid Variables for the -cmd Argument of archive_do

Variable	Description
%extract_plan	Plan file to extract at a given timestamp.
%extract_summary	Summary file to extract at a given timestamp.
%extract_user{FILENAME}	User file FILENAME to extract at a given timestamp.

Table 4-10 Valid Variables for the -cmd Argument of archive_do

Variable	Description
%insert_plan	Plan file to insert at a given timestamp.
%insert_summary	Summary file to insert at a given timestamp.
%insert_user{FILENAME}	User file FILENAME to insert at a given timestamp.
%timestamp	Current UTC timestamp, YYMMDD_HHMM.
%extract_previous_plan	Plan file extracted at previous timestamp; if this is first timestamp in archive, then the entire command is skipped for this timestamp.
%cariden_bin	Location of the binary files; this is useful if there is no path set. **Example:** %cariden_bin/table_extract

Example: This shows how to update the summary file for every plan in an archive.

```
archive_do -archive /opt/archives/my_archive -cmd "table_extract -plan-file %extract_plan
-out-file temp.txt; table_extract -plan-file %extract_previous_plan -out-file
temp_previous.txt; mate_summary -table-file temp.txt -old-table-file temp_previous.txt
-summary-format-file new_format_file.txt -out-file %insert_summary”
```

Collecting NetFlow Data

WAE can collect and aggregate exported NetFlow and related flow measurements. These measurements can be used to construct accurate demand traffic data for WAE Design and WAE Live. Flow collection provides an alternative to the estimation of demand traffic from interfaces, LSPs, and other statistics using Demand Deduction. NetFlow gathers information about the traffic flow and helps to build traffic and demand matrix. Importing flow measurements is particularly useful when there is full or nearly full flow coverage of a network's edge routers. Additionally, it is beneficial when accuracy of individual demands between external autonomous systems (ASes) is of interest, such as when tracking demands over time in WAE Live.

Network data collected separately by WAE Collector, including topology, BGP neighbors, and interface statistics, is combined with the flow measurements to scale flows and provide a complete demand mesh between both external autonomous systems and internal nodes.

WAE Collector gathers the following types of data to build a network model with flows and their traffic measurements aggregated over time and space:

- Flow traffic using NetFlow, JFlow, CFlowd, IPFIX, and Netstream flows
- Interface traffic and BGP peers over SNMP
- BGP path attributes over peering sessions

NetFlow Collection Architectures

There are two types of flow collection architectures:

- Centralized NetFlow (CNF)—Typically used for small to medium networks. This is a single-server architecture. Figure 5-1 shows the CNF workflow.

  Note Prior to WAE 6.4.9, CNF was the only architecture available.

- Distributed NetFlow (DNF)—Typically used for larger networks. This architecture consists of a JMS broker, master, and agents. Figure 5-2 shows the DNF architecture and Figure 5-3 shows the DNF workflow.

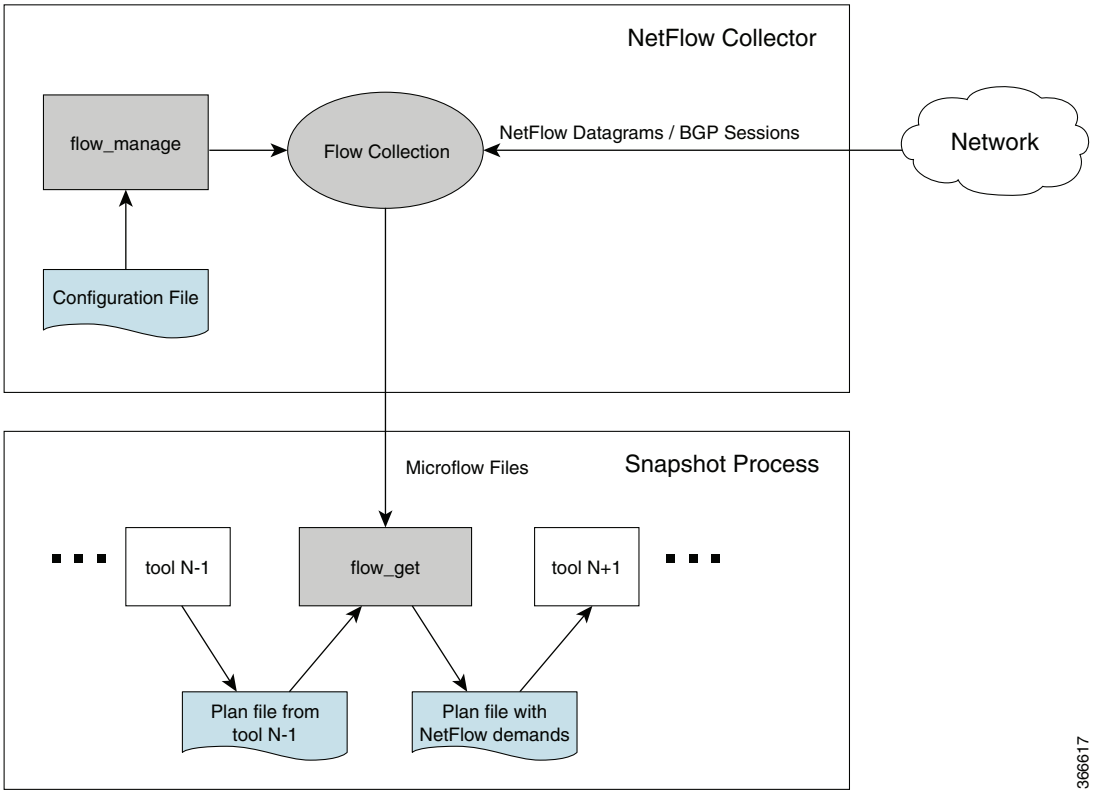
  Note The DNF architecture is only available in WAE 6.4.9 and later 6.4.x releases.

The collection architecture to deploy depends on the measured or estimated rate of NetFlow traffic export from the network in Mbps or fps.

NetFlow Collection Workflows

Figure 5-1 shows the workflow for collecting and computing flow data in CNF. The WAE Collector tools, `flow_manage` and `flow_get`, integrate with an external configuration file and the snapshot process, respectively. The end result is a plan file that contains flow-based demands and demand traffic.

Figure 5-1 Centralized Collection and Demand Creation



- `flow_manage`—This CLI tool configures network connectivity and manages the collection server, including starting, stopping and configuring the flow collection process. It uses input from the <NodeFlowConfigs> table from a configuration file to generate configuration information, which it then sends to the flow collection server. The `flow_manage` tool must be invoked outside the snapshot process. See Snapshot Integration and the `flow_manage -help` output.

Note Do not use `+s` (for sudo user) when issuing any `flow_manage` command.

- Flow collection server—This background process receives configuration information from `flow_manage`, which it uses to configure the collection server and receive flow data and BGP attributes. The collection server then aggregates this data and forwards the microflows file to the `flow_get` tool.

Note The flow collection server utilizes pmacct (a set of network monitoring tools). For more information on pmacct, see http://www.pmacct.net/.

The flow collection server waits for the receipt of a BGP OPEN message from a remote AS to establish an iBGP peering session. It acts as a passive BGP peer that only receives full BGP routing information. The flow collection server is capable of forming authenticated BGP sessions, if required.

  - The installation process sets the file capabilities for all binaries in `$CARIDEN_HOME/lib/ext/pmacct/sbin`, which enables you to collect flow data using `flow_manage` and `flow_get` without having to change the file capabilities for the flow collection server. No further configuration is needed.
  - If receiving BGP routes, the maximum length of the BGP `AS_path` attribute is limited to three hops. The reason is to prevent excessive server memory consumption, considering that the total length of BGP attributes, including `AS_path`, attached to a single IP prefix can be very large (up to 64 KB).

- `flow_get`—This CLI tool is configured inside the snapshot files. It reads flow data (microflows file) from the collection server, produces NetFlow demands and demand traffic data, and inserts this data into a plan file. In addition to producing demand and traffic data, `flow_get` also produces inter-AS (IAS) flow files. See Configure flow_get and `flow_get -help` output.

Figure 5-2 shows the DNF architecture and Figure 5-3 shows the DNF workflow. In this architecture, each set of network devices exports flow data to a corresponding collection server. The DNF cluster performs flow computation so that it is no longer performed as part of the snapshot process. Instead, each agent is responsible for the flow computation of its corresponding flow collection server that runs flow collector. The master node aggregates this information and passes it back to `flow_collector_ias`.

Figure 5-2 Distributed NetFlow Collection

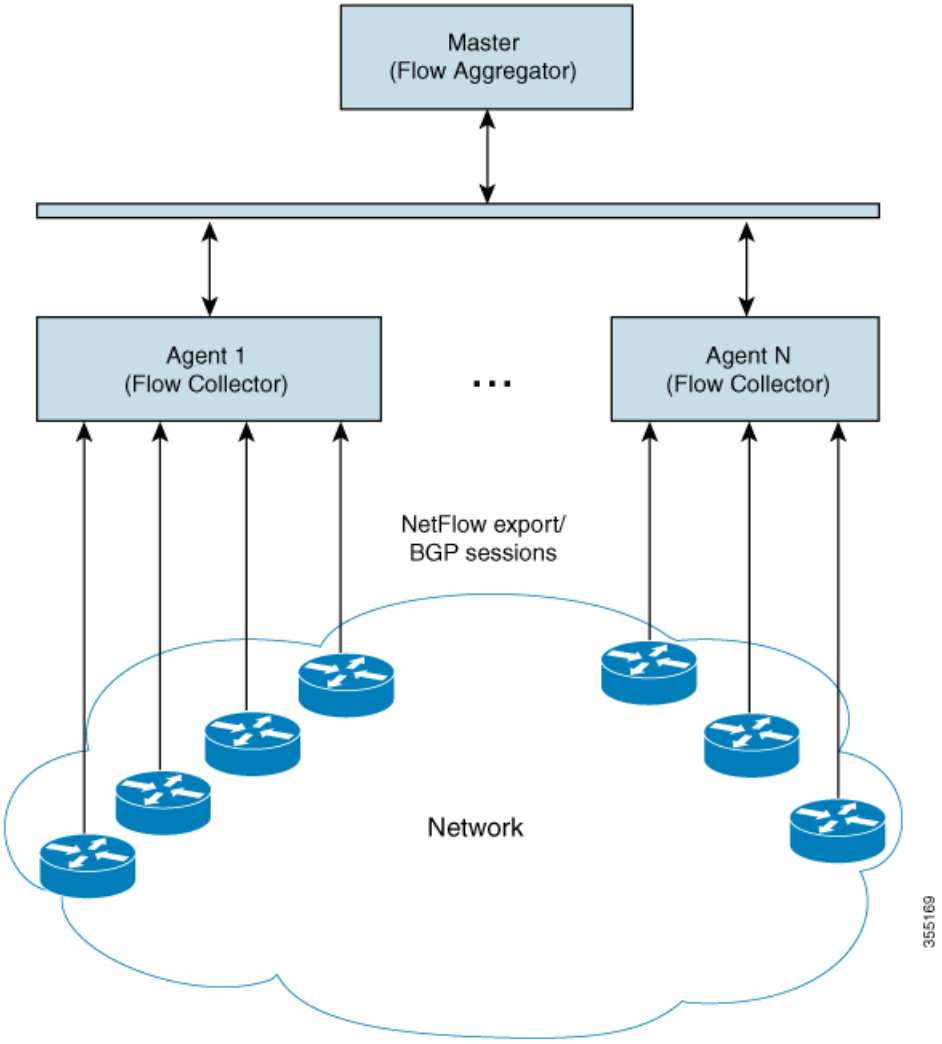
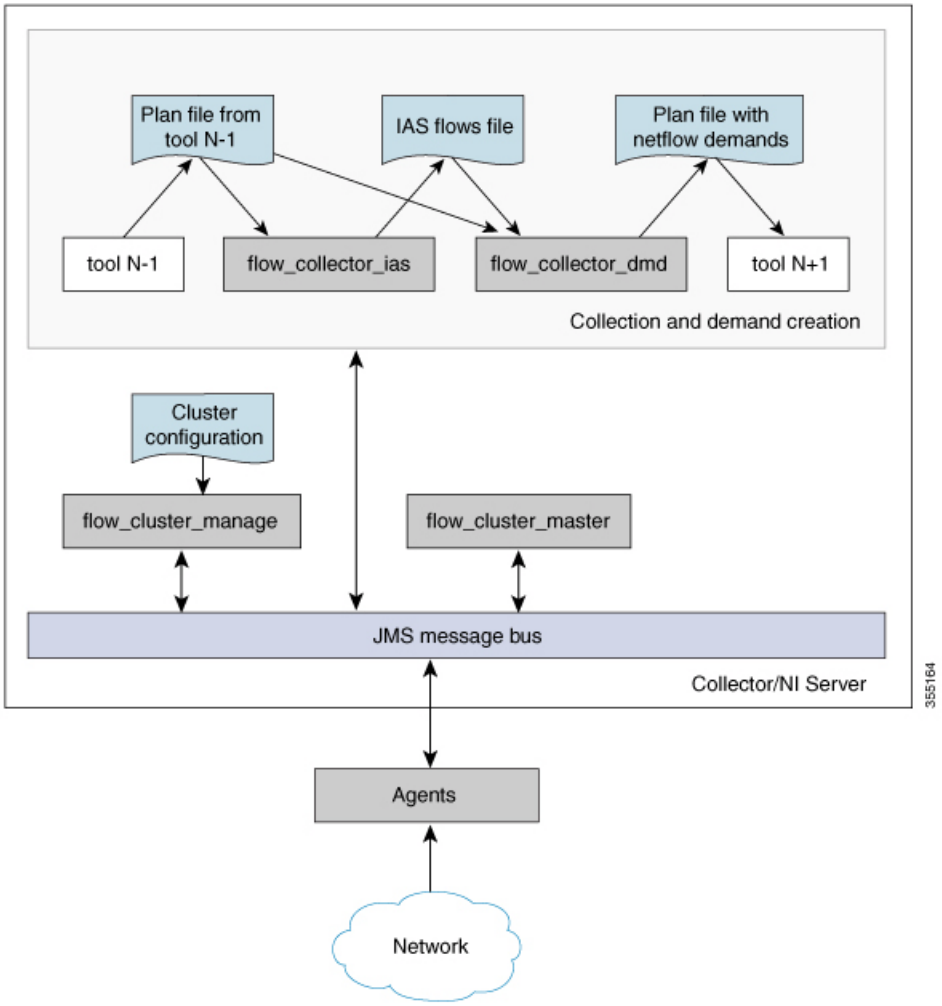


Figure 5-3 Distributed Collection and Demand Creation



- `flow_cluster_manage`—This CLI tool is used to configure and get status from the cluster. It takes a cluster configuration file and sends the configuration to the cluster. See Send the Configuration File to the Cluster.

Note Do not use `+s` (for sudo user) when issuing any `flow_cluster_manage` command.

A REST API is also available to configure and request status from the cluster as an alternative to using `flow_cluster_manage`. For more information, see the API documentation from one of the following locations:

- $CARIDEN_HOME/docs/api/netflow/distributed-netflow-rest-api.html
- http://*<master-IP-address>*:9090/api-doc

  For example, to get the cluster configuration:

  ```
  curl -X GET http://localhost:9090/cluster-config > config-file-1
  ```

  For example, to set the cluster configuration:

  ```
  curl -X PUT http://localhost:9090/cluster-config  @config-file-2
  ```

For example, to get the cluster status:

```
curl -X GET http://localhost:9090/cluster-status > config-file-1
```

- `flow_cluster_master`—The master service collects all flow data results from all the agents and aggregates the data, which is send back to `flow_collector_ias`. For information, see Master.
- `flow_cluster_agent`—The agent service manages and tracks the status of the associated flow collector. Each agent receives and computes the flow data from its corresponding collection server. For information, see Agents.
- `flow_cluster_broker`—(not shown in diagram) The JMS broker service allows communication between all components within the architecture, including master and agents. For information, see Java Message Server (JMS) Broker.
- `flow_collector_ias`—This CLI tool, which is configured inside the snapshot file, receives the flow data from the master and produces the IAS flows file. For information, see Produce NetFlow Demands.
- `flow_collector_dmd`—This CLI tool produces a plan file that only includes NetFlow demands and demand traffic as part of the snapshot process. For information, see Produce NetFlow Demands.

Note In production networks, do not use -log-level=INFO | DEBUG | TRACE for `flow_get`, `flow_collector_ias`, or `flow_collector_dmd`.

Centralized NetFlow Workflow

To configure CNF and start collection:

Step 1 Confirm that the Centralized NetFlow Requirements are met.

Step 2 Configure and Run the Collector Server.

- Edit the <NodeFlowConfigs> Table

Step 3 Configure flow_get.

- Snapshot Integration

Step 4 Configure the snapshot files to execute the appropriate snapshot tasks, including `flow_get`. See Snapshot Integration.

Step 5 Run the snapshot.

Centralized NetFlow Requirements

For system requirements, see one of the following documents:

- *Cisco WAE 6.4 - 6.4.7 System Requirements*
- *Cisco WAE 6.4.9 System Requirements*

Licensing

Confirm with your Cisco WAE representative that you have the correct licenses to obtain flow and flow demands when using the `flow_manage` and `flow_get` tools.

NetFlow Collection Configuration

The flow collection process supports IPv4 and IPv6 flows captured and exported by routers in the **ingress** direction. It also supports IPv4 and IPv6 iBGP peering.

Routers must be configured to export flows to and establish BGP peering with the flow collection server. Note the following recommendations:

- NetFlow v5, v9, and IPFIX datagram export to the UDP port number of the flow collection server, which has a default setting of 2100. Export of IPv6 flows requires NetFlow v9 or IPFIX.
- Configure the flow collection server on the routers as an iBGP route reflector client so that it can send BGP routes to edge or border routers. If this is not feasible, configure a router or route server that has a complete view of all relevant routing tables.
- Configure the source IPv4 address of flow export data grams to be the same as the source IPv4 address of iBGP messages if they are in the same network address space.
- Explicitly configure the BGP router ID.
- Configure static routing.
- If receiving BGP routes, the maximum length of the BGP `AS_path` attribute is limited to three hops. The reason is to prevent excessive server memory consumption, considering that the total length of BGP attributes, including `AS_path`, attached to a single IP prefix can be very large (up to 64 KB).

Prepare the Operating System for CNF

To prepare the OS for CNF, run the following `flow_manage` command:

```
sudo ./flow_manage -action prepare-os-for-netflow
```

The `prepare-os-for-netflow` option does the following:

- Uses the `setcap` command to allow non-root users limited access to privileged ports (0-1023). This is necessary when configuring the flow collector to use a port under 1024 to listen to BGP messages.
- Configures the OS instance to reserve up to 15,000 of file descriptors to account for the large number of temporary files that may be produced by `flow_get` in a CNF architecture.

Note After executing this command, you must reboot the server.

Configure and Run the Collector Server

The `flow_manage` tool starts and stops (`flow_manage -start` or `flow_manage -stop`) the flow collection process, as well as reloads the configuration information stored in the <NodeFlowConfigs> table when you change it. As such, you must run it before executing the snapshot process.

We recommend that you configure your operating system to automatically start and stop `flow_manage` (`-action start` or `-action stop`) at system start or shutdown.

Example: The following command reloads the <NodeFlowConfigs> table in the `flowconfigs.txt` file to a flow collection server with an IP address of 192.168.1.3. See Edit the <NodeFlowConfigs> Table for more information on creating the <NodeFlowConfigs> table.

```
flow_manage -server-ip 198.51.100.1 -action reload -node-flow-configs-table
flowconfigs.txt
```

Example 5-1 Sample Configuration File

```
<NodeFlowConfigs>
Name,BGPSourceIP,FlowSourceIP,BGPPassword,SamplingRate
ar1.dus.lab.test.com,1.2.3.4,1.2.3.5,bgp-secret,666
ar1.ham.lab.test.com,1.2.3.41,1.2.3.52,bgp-secret-2,667
cr1.ams.lab.test.com,1.2.3.51,1.2.3.53,bgp-secret-3,8000

<IPPrefixFiltering>
NetworkAddress
198.51.100.1/24
198.51.100.1/23
198.51.100.1/22
198.51.100.1/21
```

Edit the <NodeFlowConfigs> Table

The <NodeFlowConfigs> table contains basic node configuration information used by the `flow_manage` tool when generating configuration information that it passes to the flow collection server.

Thus, prior to executing `flow_manage`, you must construct this table as follows.

- Use a tab or comma delimited format.
- Include one row per node (router) from which you are collecting flow data.
- Enter contents described in Table 5-1 for each of these nodes. The BGP columns are required only if collecting BGP information. Table 5-2 provides an example.

Table 5-1 <NodeFlowConfigs> Table Columns

Column	Description
Name	Node name.
SamplingRate	Sampling rate of the packets in exported flows from the node. For example, if the value is 1,024, then one packet out of 1,024 is selected in a deterministic or random manner.
FlowSourceIP	IPv4 source address of flow export packets.
BGPSourceIP	IPv4 or IPv6 source address of iBGP update messages. This column is needed if the `flow_manage -bgp` option is true.
BGPPassword	BGP peering password for MD5 authentication. Use this column if the `flow_manage -bgp` option is true and if BGPSourceIP has a value.

Table 5-2 Example <NodeFlowConfigs> Table

Name	SamplingRate	FlowSourceIP	BGPSourceIP	BGPPassword
paris-er1-fr	1024	192.168.75.10	69.127.75.10	ag5Xh0tGbd7
chicago-cr2-us	1024	192.168.75.15	69.127.75.15	ag5Xh0tGbd7
chicago-cr2-us	1024	192.168.75.15	2001:db8:85a3::8a4e:370:7332	ag5Xh0tGbd7
tokyo-br1-jp	1024	192.168.75.25	69.127.75.25	ag5Xh0tGbd7
brazilia-er1-bra	1024	192.168.75.30	2001:db8:8:4::2	ag5Xh0tGbd7

Configure flow_get

The `flow_get` tool is executed within the snapshot process as a way to get the flow data from the flow collection server and add it to the plan file.

Example: The following command gets the data (plan file) from the previous snapshot task, adds a demand traffic matrix to it, and outputs it to the `/acme/outfile.txt` file. All external BGP interface that fail to report netflow information will be written into the /acme/ext_no.txt file. Interfaces from which flow data was received are also marked in the <NetIntInterfaces> table.

```
flow_get -plan-file /acme/infile.db -out-file /acme/outfile.db -split-as-flows-on-ingress
aggregate -demands true -missing-flows /acme/ext_no.txt
```

Example: Create a list of demands where IPv4 and IPv6 traffic is listed as separate entries in the plan file.

```
flow_get -plan-file /acme/infile.db -out-file /acme/outfile.db
-split-as-flows-on-ingress aggregate -demands true -address-family ipv4,ipv6
```

Example: Create a list of demands where IPv4 and IPv6 traffic is combined in the plan file.

Aggregated: `flow_get -plan-file /acme/infile.db -out-file /acme/outfile.db -split-as-flows-on-ingress aggregate -demands true -address-family ipv4+ipv6`

Example: Match egress IP addresses with the external addresses in the BGP peers, thus enabling you to collect flows from border routers that do not have BGP next-hop-self configured.

```
flow_get -plan-file /acme/infile.db -out-file /acme/outfile.db -split-as-flows-on-ingress
aggregate -match-on-bgp-external-info true
```

Flow Collection Perimeter

WAE Collector classifies interfaces as either internal or external. Internal interfaces are between two nodes that are in the customer network. External interfaces are those that connect a node that is discovered to one that is not. These external interfaces typically send traffic to upstream providers, downstream customers, and to peers.

The flow collection perimeter is the set of interfaces from which flow measurements are accepted, and by default, this includes external interfaces. This default definition of the flow collection perimeter might be too restrictive and could lead to discarding flow measurements on interfaces that are perceived to be internal. Following are two such examples.

- Edge devices that are hosting external interfaces that are part of the discovered topology, but do not export flows.

- Capacity planning or traffic engineering scenarios that are limited to a sub-set of the discovered network, such as to just the core network.

You can change this default flow collection perimeter by using tags to create whitelists of nodes and then passing these tags into the option `-ext-node-tags` in `flow_get`. Interfaces connected to a node matching these tags are marked as external and measurements received by these interfaces are considered.

Step 1 Tag the nodes that face the interfaces you want to be considered external to the flow collection perimeter. You can use `table_edit`, `mate_sql`, or any other CLI tool that enables you to create node tags.

Example: This example tags all Cisco devices using an IOS-XE operating system with a "non_core" tag.

```
mate_sql -file nodelist.txt -out-file core_network.txt -sql "UPDATE Nodes SET Tags =
'non_core' WHERE VENDOR = 'Cisco' AND OS = 'IOS-XE'"
```

Step 2 Send `flow_get` a list of these tags using the `-ext-node-tags` option to identify one or more comma separated tags to exclude from the flow collection perimeter.

Example: This example excludes all nodes tagged with "non_core" from the collection of flow measurements.

```
flow_get -plan-file /acme/infile.db -out-file /acme/outfile.db -split-as-flows-on-ingress
aggregate -ext-node-tags non_core
```

Snapshot Integration

The `flow_manage` tool is executed outside of snapshot files. The `flow_get` tool, however, and other necessary CLI tools are integrated within the WAE Collector snapshot process. The snapshot files include the required tasks, which must be executed in the following order. To execute a task, uncomment it (remove the initial # sign) in the snapshot.txt file. For example, the snapshot.txt file may have the following tasks configured (uncommented):

- SNMP_FIND_INTERFACES
- FIND_BGP
- SNMP_POLL
- FLOW_GET
- TRIM_NODES

Note

- While many tasks are optional, the previous sequence includes FIND_BGP and TRIM_NODES since they are commonly used.
- If the snapshot includes `collector_getplan`, then `flow_get` must be executed right after `collector_getplan`.

For more information on snapshots, see Snapshot Files.

Flow Collection Server Log Files

Use `flow_manage` to produce log files.

For example:

```
# flow_manage -log-file <filename>
```

To set the log level:

```
# flow_manage -log-level <value>
```

where *value* can be one of the following: off, activity, fatal, error, warn, notice, info, debug, or trace

To recover debugging information and produce a zip file containing all relevant pmacct configuration and log files:

```
# flow_manage -action produce-debug-file -log-level info
```

Distributed NetFlow Workflow

Note The DNF architecture is only available in WAE 6.4.9 and later 6.4.x releases.

To configure DNF and start collection:

Step 1 Confirm that the Distributed NetFlow Requirements are met.

Step 2 Configure the DNF Cluster Environment.

Step 3 Create the Cluster Configuration File.

Step 4 Send the Configuration File to the Cluster.

Step 5 Produce NetFlow Demands.

Step 6 Run the snapshot. See Snapshot Integration.

Distributed NetFlow Requirements

For system requirements, see the *Cisco WAE 6.4.9 System Requirements* document.

Licensing

Confirm with your Cisco WAE representative that you have the correct licenses for getting flow and flow demands when using the `flow_cluster_master`, `flow_collector_ias`, and `flow_collector_dmd` tools.

Java Message Server (JMS) Broker

Each distributed flow collection setup must have a single JMS broker instance in order for the master, agents, and client within a cluster to exchange information. All information is interchanged through the broker and enables all the components to communicate with each other. DNF supports a dedicated JMS broker.

The broker must have the following features enabled in order for all JMS clients (master, agents, and flow_collector_ias instances) to work:

- Out of band file messaging

- Support of obfuscated passwords in configuration files

Master and Agents

Master

The master node provides the following services in the cluster:

- Monitors and tracks agent status.
- Monitors and tracks the status of the last completed IAS computation.
- Aggregates IAS flow data coming from all agents back to the client.
- Handles configuration and status requests from the cluster.

Agents

Only one agent per server is supported. Agents cannot be on the WAE installation or snapshot server. Each agent receives and computes flow data from its corresponding collection server.

Note You have the option to deploy only one agent in the cluster. This is an alternative to CNF for networks that are expected to expand in size or grow in traffic.

Ansible files are used to install and run DNF configuration on the agent servers.

Configure the DNF Cluster Environment

Before You Begin

- You must have Ansible 1.9 installed on your installation server.
- A sudo SSH user with the same name in each server dedicated for the cluster (broker, master, and all the agents) must exist. Make a note of this username because it is used in the `group_vars/all` Ansible file (discussed later in this section).
- WAE Planning software must be installed on a server (installation server) with the appropriate license file.
- Agent system requirements meet the same requirements needed for WAE installation.
- The flow collection process supports IPv4 and IPv6 flows captured and exported by routers in the ingress direction. It also supports IPv4 and IPv6 iBGP peering. Routers must be configured to export flows to and establish BGP peering with the flow collection server. For more information, see NetFlow Collection Configuration, page 5-7.

Modify the DNF Configuration Files

If you use default WAE installation options, there are only a few mandatory parameters that must be changed. These will be noted in the applicable configuration topics. The topics described in this section assume the following:

- The master server (installation server) is where the WAE planning software has been installed and default directories are used. In particular, the configuration files used for DNF on the installation server are located in `/opt/cariden/software/mate/current/etc/netflow/ansible`.

- A dedicated JMS broker will be used in DNF configuration.
- In configuration examples, the following values are used:
  - Master and JMS broker IP address—198.51.100.10
  - Agent 1 IP address—198.51.100.1
  - Agent 2 IP address—198.51.100.2
  - Agent 3 IP address—198.51.100.3

group_vars/all

The file is located in `/opt/cariden/software/mate/current/etc/netflow/ansible/group_vars/all`. This file is the Ansible file that contains the variable definitions that are used in the playbook files.

Edit the following options:

- LOCAL_WAE_INSTALLATION_DIR_NAME—The local path that contains the WAE 6.4.x installation file.
- WAE_INSTALLATION_FILE_NAME—The filename of the WAE 6.4.x installation file.
- TARGET_JDK_OR_JRE_HOME—The full path and filename of the Oracle JRE file. All machines in the cluster (broker, master, and all the agents) should have the JRE previously installed under this variable.
- LOCAL_LICENSE_FILE_PATH—The full path to the license file.
- SSH_USER_NAME—The SSH username created or used when SSH was enabled on each machine. This sudo user is used by Ansible to deploy the cluster over SSH.

For example (comments are removed):

```
LOCAL_WAE_INSTALLATION_DIR_NAME: "/wae/wae-installation"
WAE_INSTALLATION_FILE_NAME: "wae-linux-v16.4.9-1396-g6114ffa.rpm"

TARGET_JDK_OR_JRE_HOME: "/usr/lib/jvm/java-1.8.0-openjdk-1.8.0_45"
LOCAL_LICENSE_FILE_PATH: "/home/user1/.cariden/etc/MATE_Floating.lic"

TARGET_SSH_USER: ssh_user
```

hosts

The file is located in `/opt/cariden/software/mate/current/etc/netflow/ansible/hosts`. This file is the Ansible inventory file and it includes a list of all the servers in the cluster.

Only edit the corresponding IP addresses for the broker, master, and all agents. Do not edit any of the other variables. If applicable, add more agents.

For example:

```
[dnf-broker]
198.51.100.10 ansible_ssh_user={{SSH_USER_NAME}}

[dnf-master]
198.51.100.10 ansible_ssh_user={{SSH_USER_NAME}}

[dnf-agent-1]
198.51.100.1 ansible_ssh_user={{SSH_USER_NAME}}

[dnf-agent-2]
198.51.100.2 ansible_ssh_user={{SSH_USER_NAME}}
```

```
[dnf-agent-3]
198.51.100.3 ansible_ssh_user={{SSH_USER_NAME}}
```

prepare-agents.yml

This file does not need to be edited and provides the following to all specified agents:

- Allows non-root users limited access to privileged ports (0-1023). This is necessary when configuring the flow collector to use a port under 1024 to listen to BGP messages.
- Configures the OS instance to reserve up to 15,000 of file descriptors to account for the large number of temporary files that may be produced by `flow_get` in a CNF architecture.
- Reboots all the agents.

The file is located in `/opt/cariden/software/mate/current/etc/netflow/ansible/prepare-agents.yml`.

startup.yml

The file is located in `/opt/cariden/software/mate/current/etc/netflow/ansible/startup.yml`.

This file is used to automatically start the broker, master, and agents. If you have more than two agents, edit this file to add more.

For example:

```
- hosts: all
  roles:
     - check-ansible-version

- hosts: dnf-broker
  roles:
     - start-broker

- hosts: dnf-master
  roles:
     - start-master

- hosts: dnf-agent-1
  roles:
     - {role: start-agent, instance: instance-1}

- hosts: dnf-agent-2
  roles:
     - {role: start-agent, instance: instance-2}

- hosts: dnf-agent-3
  roles:
     - {role: start-agent, instance: instance-3}
```

service.conf

The file is located in `/opt/cariden/software/mate/current/etc/netflow/ansible/bash/service.conf`.

This file provides the common configuration options that are used by the broker, master, and agents.

Edit the following options:

- jms-broker-server-name-or-ip-address—IP address of the broker.
- jms-broker-jms-port—JMS port number being used for the broker.

- jms-broker-http-port—HTTP port number being used for the broker.
- jms-broker-username—This is used internally and does not need to be changed.
- jms-broker-password—We recommend generating and using an obfuscated password. For example:

```
# ./flow_cluster_manage -action print-obfuscation
type in the clear text > password-0
obfuscated text: obfuscated text: ENC(h4rWRpG54WgVZRTE90Zb/JszY4dd4CGc)
```

- obfuscated text: ENC(h4rWRpG54WgVZRTE90Zb/JszY4dd4CGc)
- jms-broker-use-tls—To encrypt all data communication in the DFC cluster, then enter true. If set to true, there will be some performance degradation.
- append-to-log-file—If appending information to the local log file, enter true.
- use-flume—If using a flume server, enter true.
- flume-server—Enter the IP address of the server running the flume agent. If using the flume server that is automatically installed during WAE server installation, enter the installation server IP address.
- log-level—Enter logging level type:
  - off
  - activity
  - fatal
  - error
  - warn
  - notice
  - info
  - debug
  - trace

For example:

```
# jms
jms-broker-server-name-or-ip-address=198.51.100.10
jms-broker-jms-port=61616
jms-broker-http-port=8161
jms-broker-username=user-0
jms-broker-password=ENC(ctrG7GGRJm983M0AsPGnabwh)
jms-broker-use-tls=false

# local logging
append-to-log-file=false

# distributed logging
use-flume=true
flume-server=198.51.100.10

# default for all commands, will be superseded if specified locally in each .sh
log-level=info
```

Deploy DNF Cluster

To deploy the DNF cluster:

Step 1 Install the broker, master and agents:

```
# ansible-playbook -i hosts install.yml
```

Note The `uninstall.yml` playbook file uninstalls the files and removes the TARGET_WAE_ROOT directory, which is defined in the `all` file.

Step 2 Prepare and reboot the agents for DNF:

```
# ansible-playbook -i hosts prepare-agents
```

Step 3 Start the master, broker, and agents.:

```
# ansible-playbook -i hosts startup.yml
```

Note The `shutdown.yml` playbook file shuts down the master, broker, and agents.

Step 4 Confirm that the master, broker, and agents are running:

```
# ansible-playbook -i hosts list.yml
```

Step 5 After the machines reboot, you can verify if all the agents are up by executing the following command:

```
flow_cluster_manage -active request-cluster-status
```

A successful result should list running details of the master and all agents. At the end of the result, the CLUSTER SUMMARY should look similar to the following:

```
CLUSTER SUMMARY - BEGIN
        cluster all OK:          false
        configured size:         0
        agents up:               2
        daemons up:              0
        agents w/wrong IDs:      []
        agents w/low ulimit IDs: []
        computation mode:        ias-in-the-background
        last result time:        n/a
        last no-result time:     n/a
        max diff time:           2 ms
        max diff time OK:        true
CLUSTER SUMMARY - END
```

Note that in the preceding example, the `agents up` lists two running agents. The `cluster all OK` field is false because the cluster has not been configured yet. This status should change to true after running through all the Ansible playbooks.

Create the Cluster Configuration File

To more easily create the cluster configuration file for `flow_cluster_manage`, you can use the configuration file produced from `flow_manage` as a template for the cluster configuration file.

For example:

1. Produce the template configuration file:

```
${CARIDEN_HOME}/flow_manage \
-action produce-config-file \
-node-flow-configs-table <input-path> \
-cluster-config-file <output-path> \
-interval 120 \
-bgp true \
-bgp-port 10179 \
-port 12100 \
-flow-size lab \
-server-ip ::
```

where <*input-path*> is the path of the node configuration .txt file used in CNF (see Configure and Run the Collector Server for more information on creating this file) and <*output-path*> is the path where you want the resulting seed cluster configuration file to reside. Verify that the output of the seed cluster configuration file is similar to the following:

```
{
    "agentConfigMapInfo": {
        "cluster_1::instance_1":
        {
            "flowManageConfiguration":
            {
                "maxBgpdPeers": 150,
                "bgpTcpPort": 179,
                "flowType": "Netflow",
                "useBgpPeering": true,
                "outfileProductionIntervalInSecs": 900,
                "networkDeploymentSize": "medium",
                "netflowUdpPort": 2100,
                "keepDaemonFilesOnStartStop": true,
                "purgeOutputFilesToKeep": 3,
                "daemonOutputFileMaskSuffix": "%Y.%m.%d.%H.%M.%s",
                "daemonOutputDirPath":
"<user.home>/.cariden/etc/net_flow/flow_matrix_interchange",
                "daemonOutputFileMaskPrefix": "out_matrix_",
                "daemonOutputSoftLinkName": "flow_matrix_file-latest",
                "extraAggregation": [],
                "routerConfigList":
                    [
                        {
                            "name": "ar1.dus.lab.cariden.com",
                            "bGPSourceIP": "1.2.3.4",
                            "flowSourceIP": "1.2.3.5",
                            "bGPPassword": "bgp-secret",
                            "samplingRate": "666"
                        },
                        {
                            "name": "cr1.ams.lab.cariden.com",
                            "bGPSourceIP": "1.2.3.51",
                            "flowSourceIP": "1.2.3.53",
                            "bGPPassword": "bgp-secret-3",
                            "samplingRate": "8000"
                        }
                    ],
                "appendedProperties":
                {
                    "key1": "value1",
                    "key2": "value2"
                }
            }
        },
}
```

2. Edit the file to include each agent configuration. Copy, paste, and edit each section as it applies to each agent in the cluster. This example shows two agents:

```
{
    "agentConfigMapInfo": {
        "cluster_1::instance_1":
        {
            "flowManageConfiguration":
            {
                "maxBgpdPeers": 150,
                "bgpTcpPort": 179,
                "flowType": "Netflow",
                "useBgpPeering": true,
                "outfileProductionIntervalInSecs": 900,
                "networkDeploymentSize": "medium",
                "netflowUdpPort": 2100,
                "keepDaemonFilesOnStartStop": true,
                "purgeOutputFilesToKeep": 3,
                "daemonOutputFileMaskSuffix": "%Y.%m.%d.%H.%M.%s",
                "daemonOutputDirPath":
"<user.home>/.cariden/etc/net_flow/flow_matrix_interchange",
                "daemonOutputFileMaskPrefix": "out_matrix_",
                "daemonOutputSoftLinkName": "flow_matrix_file-latest",
                "extraAggregation": [],
                "routerConfigList":
                    [
                        {
                            "name": "ar1.dus.lab.anyname.com",
                            "bGPSourceIP": "1.2.3.4",
                            "flowSourceIP": "1.2.3.5",
                            "bGPPassword": "bgp-secret",
                            "samplingRate": "666"
                        },
                        {
                            "name": "cr1.ams.lab.anyname.com",
                            "bGPSourceIP": "1.2.3.51",
                            "flowSourceIP": "1.2.3.53",
                            "bGPPassword": "bgp-secret-3",
                            "samplingRate": "8000"
                        }
                    ],
                "appendedProperties":
                {
                    "key1": "value1",
                    "key2": "value2"
                }
            }
        },
```

The information for the second agent starts here:

```
{
        "cluster_1::instance_2":
        {
            "flowManageConfiguration":
            {
                "maxBgpdPeers": 150,
                "bgpTcpPort": 179,
                "flowType": "Netflow",
                "useBgpPeering": true,
                "outfileProductionIntervalInSecs": 900,
                "networkDeploymentSize": "medium",
                "netflowUdpPort": 2100,
```

```
                "keepDaemonFilesOnStartStop": true,
                "purgeOutputFilesToKeep": 3,
                "daemonOutputFileMaskSuffix": "%Y.%m.%d.%H.%M.%s",
                "daemonOutputDirPath":
"<user.home>/.cariden/etc/net_flow/flow_matrix_interchange",
                "daemonOutputFileMaskPrefix": "out_matrix_",
                "daemonOutputSoftLinkName": "flow_matrix_file-latest",
                "extraAggregation": [],
                "routerConfigList":
                    [
                        {
                            "name": "ar1.dus.lab.anyname.com",
                            "bGPSourceIP": "5.6.7.8",
                            "flowSourceIP": "5.6.7.9",
                            "bGPPassword": "bgp-secret-2",
                            "samplingRate": "666"
                        },
                        {
                            "name": "cr1.ams.lab.anyname.com",
                            "bGPSourceIP": "5.6.7.81",
                            "flowSourceIP": "5.6.7.83",
                            "bGPPassword": "bgp-secret-4",
                            "samplingRate": "8000"
                        }
                    ],
                "appendedProperties":
                {
                    "key1": "value1",
                    "key2": "value2"
                }
            }
        }

}
```

Send the Configuration File to the Cluster

The `flow_cluster_manage` tool diagnoses and controls the distributed NetFlow collection cluster. After creating the configuration file, use `flow_cluster_manage` to send the cluster configuration file to the cluster (`flow_cluster_manage -send-cluster-configuration`). All flow collection processes in all agents will reload the configuration information stored in that configuration file.

Note

- We recommend that you configure your system to automatically start and stop `flow_cluster_master, flow_cluster_agent,` and `flow_cluster_broker` at system start or shutdown.
-

You can also use the flow_cluster_manage tool to retrieve cluster status. For example:

```
flow_cluster_manage -action request-cluster-status
```

Note The cluster will take approximately a minute to take the configuration.

Example 5-2 Sample result of cluster status

```
CLUSTER STATUS - BEGIN

    AGENT NODE - BEGIN
        cluster ID:             cluster_1
        instance ID:            instance_1
        process ID:             15292
        start time:             2017-07-10.09:19:43.000-0700
        up time:                00d 00h 00m 40s 824ms
        unique ID:
bc.30.5b.df.8e.b5-15292-1729199940-1499703582925-1a23cb00-ed76-4861-94f5-461dcd5b2070
        last HB received:       2017-07-10.09:20:24.004-0700
        last HB age:            00d 00h 00m 04s 779ms
        skew time:              00d 00h 00m 00s 010ms
        computation sequence    0
        computational model     ias-in-the-background
        computing IAS:          false
        ip addresses:           [128.107.147.112, 172.17.0.1,
2001:420:30d:1320:24a8:5435:2ed5:29ae, 2001:420:30d:1320:be30:5bff:fedf:8eb5,
2001:420:30d:1320:cd72:ec61:aac8:2e72, 2001:420:30d:1320:dc55:a772:de80:a73f]
        mac address:            bc.30.5b.df.8e.b5
        jvm memory utilization: 4116Mb/4116Mb/3643Mb
        max opened files:       15000
        processors:             8
        daemon period:          00d 00h 15m 00s 000ms
        daemon out dir:
/media/1TB/user1/sandboxes/git/netflow-flexible/package/linux-release/lib/ext/pmacct/insta
nces/flow_cluster_agent_cluster_1::instance_1
        daemon process ID:      15344
        daemon is:              running
        bgp port:               179
        bgp port status:        up
        netflow port:           2100
        netflow port status:    up
    AGENT NODE - END

    AGENT NODE - BEGIN
        cluster ID:             cluster_1
        instance ID:            instance_2
        process ID:             15352
        start time:             2017-07-10.09:19:49.000-0700
        up time:                00d 00h 00m 30s 748ms
        unique ID:
bc.30.5b.df.8e.b5-15352-1729199940-1499703589727-12989336-b314-4f85-9978-242882dd16da
        last HB received:       2017-07-10.09:20:20.746-0700
        last HB age:            00d 00h 00m 08s 037ms
        skew time:              00d 00h 00m 00s 014ms
        computation sequence    0
        computational model     ias-in-the-background
        computing IAS:          false
        ip addresses:           [128.107.147.112, 172.17.0.1,
2001:420:30d:1320:24a8:5435:2ed5:29ae, 2001:420:30d:1320:be30:5bff:fedf:8eb5,
2001:420:30d:1320:cd72:ec61:aac8:2e72, 2001:420:30d:1320:dc55:a772:de80:a73f]
        mac address:            bc.30.5b.df.8e.b5
        jvm memory utilization: 4116Mb/4116Mb/3643Mb
        max opened files:       15000
        processors:             8
        daemon period:          00d 00h 15m 00s 000ms
        daemon out dir:
/media/1TB/user1/sandboxes/git/netflow-flexible/package/linux-release/lib/ext/pmacct/insta
nces/flow_cluster_agent_cluster_1::instance_2
        daemon process ID:      15414
        daemon is:              running
```

```
        bgp port:               10179
        bgp port status:        up
        netflow port:           12100
        netflow port status:    up
    AGENT NODE - END

    MASTER NODE - BEGIN
        cluster ID:             cluster_1
        instance ID:            instance_id_master_unique
        process ID:             15243
        start time:             2017-07-10.09:19:34.000-0700
        up time:                00d 00h 00m 50s 782ms
        unique ID:
bc.30.5b.df.8e.b5-15243-415138788-1499703574719-cd420a81-f74c-49d4-a216-ffeb7cde31d5
        last HB received:       2017-07-10.09:20:25.563-0700
        last HB age:            00d 00h 00m 03s 220ms
        ip addresses:           [128.107.147.112, 172.17.0.1,
2001:420:30d:1320:24a8:5435:2ed5:29ae, 2001:420:30d:1320:be30:5bff:fedf:8eb5,
2001:420:30d:1320:cd72:ec61:aac8:2e72, 2001:420:30d:1320:dc55:a772:de80:a73f]
        mac address:            bc.30.5b.df.8e.b5
        jvm memory utilization: 2058Mb/2058Mb/1735Mb
        processors:             8
    MASTER NODE - END

    CLUSTER SUMMARY - BEGIN
        cluster all OK:          true
        configured size:         2
        agents up:               2
        daemons up:              2
        agents w/wrong IDs:      []
        agents w/low ulimit IDs: []
        computation mode:        ias-in-the-background
        last result time:        n/a
        last no-result time:     n/a
        max diff time:           4 ms
        max diff time OK:        true
    CLUSTER SUMMARY - END

CLUSTER STATUS - END
```

The CLUSTER SUMMARY entry at the end of the result gives you a quick summary of whether or not your cluster configuration is operational. You should confirm that `cluster all OK` is true and that the `configured size`, `agents up`, and `daemons up` match the number of agents you configured. There should be no value in `agents w/wrong IDs` and `agents w/low ulimit IDs`. The `max diff time OK` should also be set to true. If this is not the case, look into the agent and master details for troubleshooting information.

Produce NetFlow Demands

The `flow_collector_ias` and `flow_collector_dmd` tools generate demands and demand traffic into a plan file with NetFlow data received from the cluster.

The `flow_collector_ias` tool reads a plan file and produces an IAS flows data file.

Example: The following command gets the data from the `/acme/infile.db` file, adds aggregated traffic from all ASNs, and produces the IAS flows file in `/acme/outfile.txt`.

```
flow_collector_ias -plan-file /acme/infile.db -split-as-flows-on-ingress aggregate
inter-as-flows-file /acme/ias_outfile.txt
```

The `flow_collector_dmd` tool reads from both a plan file and an IAS flows data file and produces a plan file with NetFlow demands.

Example: Create a list of demands where IPv4 and IPv6 traffic is separated in the plan file.

```
flow_collector_dmd -plan-file /acme/infile.db -out-file /acme/outfile.db
-split-as-flows-on-ingress aggregate true -address-family ipv4,ipv6
```

Example: Create a list of demands where IPv4 and IPv6 traffic is combined in the plan file.

```
flow_collector_dmd -plan-file /acme/infile.db -out-file /acme/outfile.db
-split-as-flows-on-ingress aggregate true -address-family ipv4+ipv6
```

Example: Match egress IP addresses with the external addresses in the BGP peers, thus enabling you to collect flows from border routers that do not have BGP next-hop-self configured.

```
flow_collector_dmd -plan-file /acme/infile.db -out-file /acme/outfile.db
-split-as-flows-on-ingress aggregate -match-on-bgp-external-info true
```

Flow Collection Perimeter

See Flow Collection Perimeter, page 5-9, which also applies to DNF.

Snapshot Integration

The `flow_cluster_manage` tool is executed outside of snapshot files. The `flow_collector_ias` and `flow_collector_dmd` tools, however, and other necessary CLI tools are integrated within the snapshot process. The snapshot files include the required tasks, which must be executed in the following order. To execute a task, uncomment it (remove the initial # sign).

For example, the snapshot.txt file may have the following tasks configured (uncommented):

- SNMP_FIND_INTERFACES
- FIND_BGP
- SNMP_POLL
- FLOW_COLLECTOR_IAS
- FLOW_COLLECTOR_DMD
- TRIM_NODES

Note

- While many tasks are optional, the previous sequence includes FIND_BGP and TRIM_NODES because they are commonly used.
- If the snapshot includes `collector_getplan`, then `flow_collector_ias` and `flow_collector_dmd` must be executed right after `collector_getplan`.

For more information on snapshots, see Snapshot Files.

Configuring Multi-Layer Network Collection

Multi-layer (L1 and L3) network collection is an advanced collection configuration. This section describes how to configure inventory, topology, and traffic collection from a multi-layer network. After installing the multi-layer package, you are able to collect and model the following information:

- Topology from DWDM networks that support Generalized Multiprotocol Label Switching (GMPLS) with non-User Network Interface (UNI) circuits
- Dynamic L1 circuit paths
- Unprotected and restorable paths
- Actual L1 circuit path hops
- Feasibility metrics and limit
- Inactive L1 links

After collection, the network model (plan file) is placed in the Design Archive. You can open the plan file and view L1 and L3 topology in WAE Design. For more information, see "Layer 1 Simulation" in the *Cisco WAE Design User Guide*.

Prerequisites

- Obtain the YANG runtime software from your Cisco WAE representative before proceeding with multi-layer configuration. The YANG runtime software is required to collect data from the optical network.
- Confirm that all system requirements for multi-layer collection have been met with a Cisco WAE representative.
- Confirm that the redhat-lsb-core and perl-Params-Check packages are installed. If you do not have the packages, install them. For example, on CentOS:

```
$ sudo yum install -y redhat-lsb-core
$ sudo yum install -y perl-Params-Check
```

- Confirm you have the seed node credentials for discovery.
- Create an authentication file by running the `mate_auth_init` CLI tool. After the authentication file (auth.enc) is created, add the following lines to the file (using tabs, not spaces):

```
<AuthServerGroup>

        AuthServerGroupName     Username     Password

        ncs     admin     admin
```

For more information on using the `mate_auth_init` tool, see Network Authentication.

Multi-Layer Configuration Workflow

The following steps describe the high-level workflow of multi-layer collection configuration.

Step 1 Download and Install the Multi-Layer (ML) Package

This step only needs to be done once.

Step 2 Start Multi-Layer Services

Step 3 Configure Layer 1 Collection

Step 4 Configure L3 Collection and Merge L1 Information in Snapshot

Step 5 (Optional) Collect from Multiple Networks

This step can be repeated depending on the number of L1 networks that need collection.

Download and Install the Multi-Layer (ML) Package

Confirm you have met all the requirements documented in the Prerequisites section.

The package installs necessary components used for multi-layer collection. See Table 6-1 for a list of services and associated command options that are installed. See Table 6-2 to see where all the components are installed.

Step 1 Log into the Planning Server with the username and password used during the WAE Planning Software installation. The default user is **wae** and the password is **ciscowae**.

Step 2 Go to the software download center where you obtained the WAE Planning Software. From there, download the WAE Optical Plug-in package <`wae-ml-collector-xxxxxx-x86_64.bin`>.

Step 3 Run the following commands:

```
$ source /etc/profile.d/mate.sh
$ sh <ML_package> -nso-install <nso_install_dir> -nso-run <nso-run directory> -wae-home
<wae-design installed directory>
```

For example:

```
$ source /etc/profile.d/mate.sh
$ sh wae-ml-collector-6.4.0-Linux-x86_64.bin -nso-run /opt/ncs-run -nso-install
/opt/ncs-wae-home /opt/cariden/software
```

Table 6-1 Multi-Layer Package Contents

Service	Description
`wae-ml`	The `wae-ml` service performs the following actions: • `start`—Starts multi-layer services. • `stop`—Stops multi-layer services. • `status`—Lists status of all multi-layer services. • `restart`—Restarts multi-layer services. • `reload`—Reloads multi-layer packages. • `cli`—Starts the service console.
`wae-optical` `wae-optical-`*n*	The `wae-optical` service performs the following actions: • `start`—Starts the plug-in instance. • `stop`—Stops the plug-in instance. • `status`—Checks the status of the plug-in instance. • `restart`—Restarts the optical plug-in instance. • `create`—Creates another optical plug-in instance under $WAE_HOME. You need to create additional optical plug-in instances when collecting multi-layer information from multiple networks. The new instance will have it's own service name as follows: wae-optical-1, wae-optical-2, and so on. See Collect from Multiple Networks for more information. • `list`—Lists all optical plug-in instances. • `delete`—Deletes the indicated optical plug-in instance. **Note** The first optical plug-in instance cannot be deleted.

Table 6-2 Directory Structure

Component	Location
Optical plug-in	WAE_HOME
Documentation	WAE_HOME/docs/wae-ml/
Controller and service scripts	WAE_ROOT/bin

Start Multi-Layer Services

Step 1 Edit and save the `$WAE_HOME/optical-plugin/config/ctc-connectors-domain.properties` file with the following information:

- `network.id`—Optical network name. For example, cisco:network
- `network.nodes.vendor`—Node vendor.
- `network.discovery.start.node`—The IP address of the discovery seed node.
- `network.discovery.start.node`—The ID to log into the seed node.
- `network.discovery.start.node.password`—The password to access the seed node.

- `network.discovery.inactivity.period`—Time in milliseconds which the discovery will time out if there is no access to the network.

Step 2 Start the ML services:

```
$ wae-ml start
```

Step 3 Confirm that the services are running:

```
$ wae-ml status
```

Configure Layer 1 Collection

You can start and configure L1 collection using the CLI console.

Step 1 Start the console:

```
$wae-ml cli
```

Step 2 Enter into configuration mode:

```
admin@ncs# config
```

Step 3 Configure the auth.enc and cariden-home paths using the Network Service Module setting options:

```
admin@ncs(config)# services NsmSettings server cariden-home <$CARIDEN_HOME>
admin@ncs(config)# services NsmSettings server cariden-auth-file <auth-file-path>
```

For example:

```
admin@ncs(config)# services NsmSettings server cariden-home /home/wae

admin@ncs(config)# services NsmSettings server cariden-auth-file
/home/wae/.cariden/etc/auth.enc
```

Step 4 Configure the L1Server IP address and optical plug-in port for the network:

```
admin@ncs(config)# services NsmSettings network <network_name> L1Server IPAddress
<system_ip_address> port <optical_plugin_port>
```

For example:

```
admin@ncs(config)# services NsmSettings network NetworkA L1Server IPAddress 172.20.162.101
port 9000
```

Note The default port is 9000 for the first optical plug-in instance. Default for the next additional optical plug-in instances are 9001, 9002, 9003, and so on. To configure the optical plug-in to connect to a different port, edit the line `restconf.http.port=<port_number>` in the `$WAE_HOME/optical-plugin[-<instance_number>]/config/ctc-connectors-restconf.properties` file.

Step 5 Schedule L1 collection:

```
admin@ncs(config)# services NsmSettings network <network_name> schedules schedule
run-l1-collection minutes <interval_for_L1_Collection_in_minutes>
```

For example:

```
admin@ncs(config)# services NsmSettings network NetworkA schedules schedule
run-l1-collection minutes 15
```

Step 6 Commit the collection and exit the console:

```
admin@ncs(config)# commit
admin@ncs(config)# exit
admin@ncs # exit
```

Configure L3 Collection and Merge L1 Information in Snapshot

Step 1 Edit the $CARIDEN_HOME/etc/snapshot.txt file:

- Enter the appropriate snapshot information, including the following parameters:
  - `network`
  - `unique`
  - `seed_router`
  - `igp`
  - `home_dir`
  - `cariden_home`
- Uncomment the tools for L3 collection. To uncomment a task, remove the # sign. For examples on what type of information to collect, see Appendix A, "Snapshot Examples".

Note To collect L1-L3 port mapping in non-UNI networks, contact your Cisco representative.

- Uncomment ACCESS_NETCONF. The `access_netconf` tool merges the L1 data with the L3 data.

Step 2 Run the snapshot collection:

```
# cd $CARIDEN_HOME/bin
# ./snapshot -config-file ../etc/snapshot.txt
```

Step 3 Open the plan files using WAE Design. Alternatively, you can collect multi-layer information from additional networks. For more information, see Collect from Multiple Networks.

Collect from Multiple Networks

The following procedure describes how to include additional L1 networks to the multi-layer collection. This procedure assumes that you have already performed a previous collection (as described earlier in this section) and the additional L1 networks are connected to a single L3 network.

Step 1 Create a new instance of an optical-plugin:

```
# wae-optical create
```

Step 2 Configure L1 collection.

Follow the same steps described in Configure Layer 1 Collection, but the network name, L1Server IP address, and port in the new network should refer to the network name IP Address and port of the new optical plug-in instance.

Step 3 Start the new instance of an optical plug-in:

```
# wae-optical-n start
```

For example:

```
# wae-optical-1 start
```

Step 4 Run the `access_netconf` tool with the new network name and old output plan as the input.

For example:

```
# ./access_netconf -network NetworkB -authGroup ncs -action merge_l1_data
 -input-plan-file post-NetworkA.txt -output-plan-file post-NetworksAB.txt
```

You can repeat this procedure to merge collections from additional networks.

Exclude Optical Amplifiers from Collection

By default, L1 collection includes amplifier link and node information. You may want to remove optical amplifier information to declutter the topology shown when opening a plan file in WAE Design. Figure 6-1 and Figure 6-2 show the differences in topology when amplifiers are included and when they are not.

To exclude optical amplifiers from collection, do the following:

Step 1 Start the ML CLI:

```
$wae-ml cli
```

Step 2 Enter into configuration mode:

```
admin@ncs# config
```

Step 3 From the Network Service Module setting options, set the `retain-amplifiers` option to false:

```
admin@ncs(config)# services NsmSettings network <network_name> options retain-amplifiers
false
```

For example:

```
admin@ncs(config)# services NsmSettings network cisco:network options retain-amplifiers
false
```

Step 4 Commit the collection and exit the console:

```
admin@ncs(config)# commit
admin@ncs(config)# exit
```

The next time L1 collection is scheduled, optical amplifier information is not included in the plan file.

Figure 6-1 Topology With Amplifiers

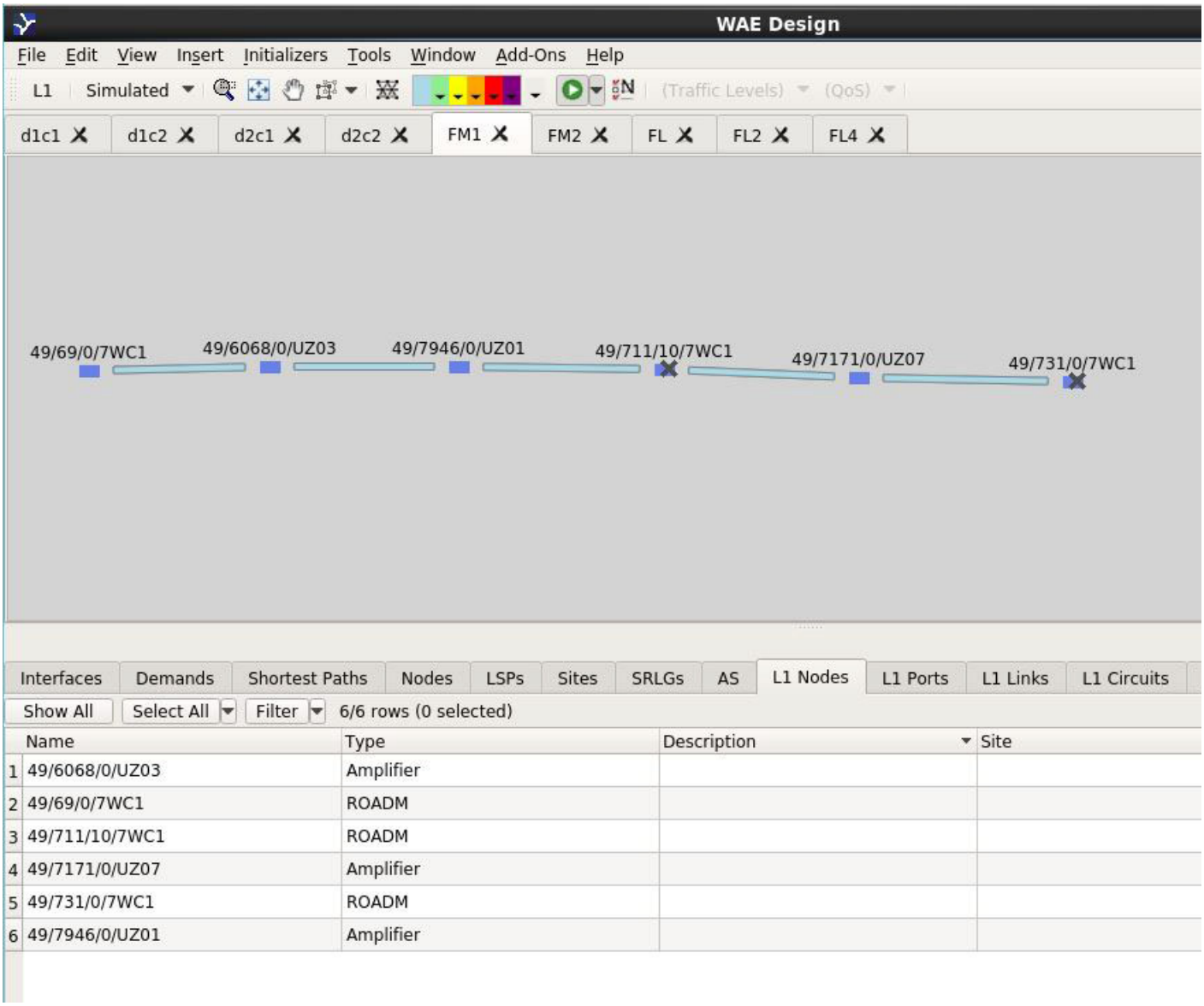
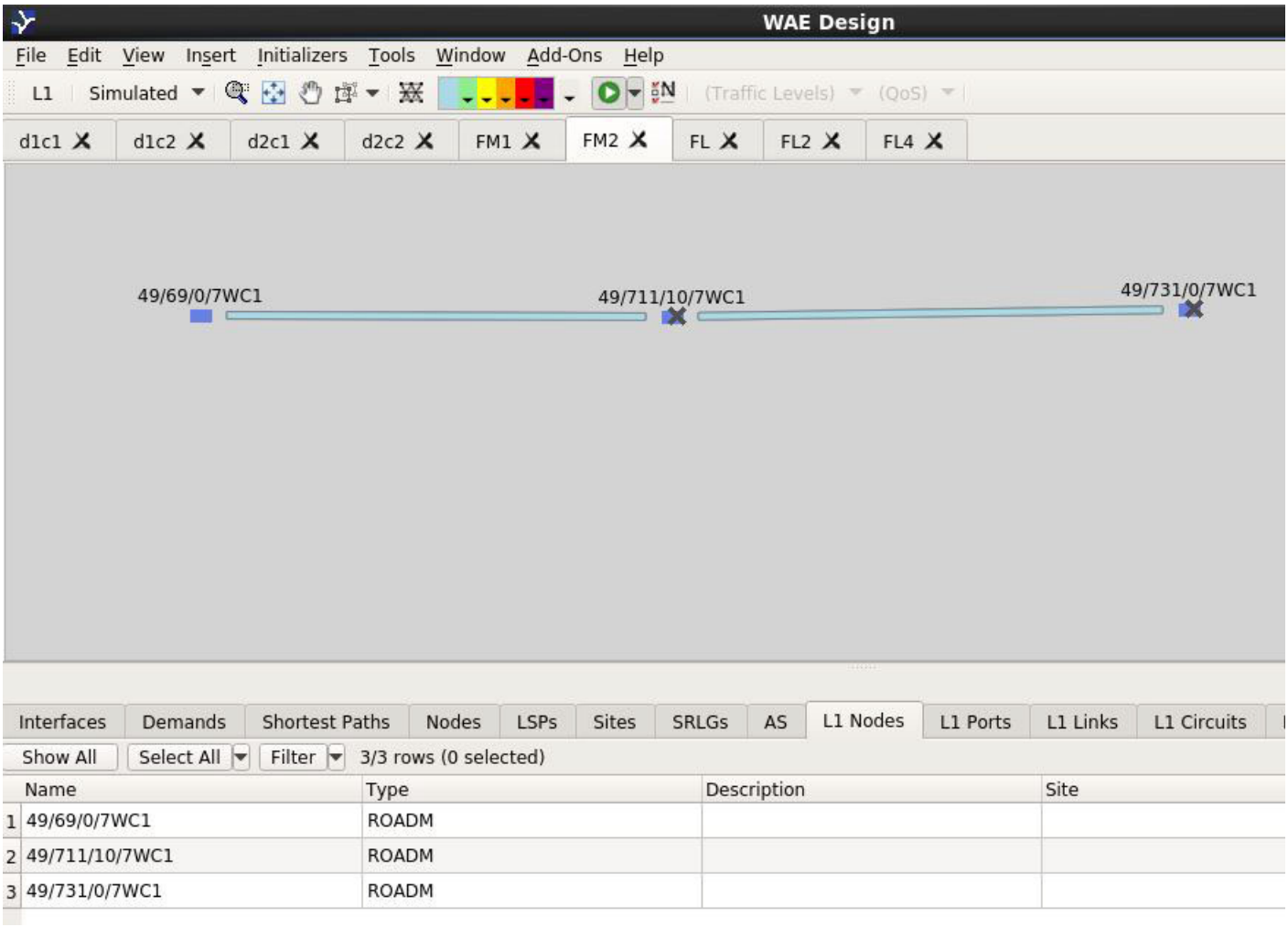


Figure 6-2 Topology Without Amplifiers



Set Feasibility Properties for L1 Circuits

The quality of an L1 circuit deteriorates as it passes through L1 links. Using feasibility properties, WAE Design enables you to simulate the weakening of L1 circuits to determine if they have degraded to the point of being unroutable. To set the feasibilty limit margin, do the following:

Step 1 Start the console:

```
$wae-ml cli
```

Step 2 Enter into configuration mode:

```
admin@ncs# config
```

Step 3 From the Network Service Module setting options, set the decimal value for the `feasibility-limit-margin`. The default value is 2.5.

```
admin@ncs(config)# services NsmSettings network <network_name> options
feasibility-limit-margin <margin_decimal_value>
```

For example:

```
admin@ncs(config)# set services NsmSettings network cisco:network options
feasibility-limit-margin 2.4
```

Step 4 Commit the collection:

```
admin@ncs(config)# commit
admin@ncs(config)# exit
```

The next time L1 collection is scheduled, the set margin is implemented in the plan file.

Collect Inactive or Failed L1 Circuit Objects

To collect failed L1 nodes or L1 links, do the following:

Step 1 Start the console:

```
$wae-ml cli
```

Step 2 Enter into configuration mode:

```
admin@ncs# config
```

Step 3 From the Network Service Module setting options, set the `compute-inactive-links` option to true:

```
admin@ncs(config)# services NsmSettings network <network_name> options
compute-inactive-links true
```

For example:

```
admin@ncs(config)# services NsmSettings network cisco:network options
impute-inactive-links true
```

Step 4 Commit the collection:

```
admin@ncs(config)# commit
admin@ncs(config)# exit
```

The next time L1 collection is scheduled, failed L1 links and L1 nodes are included in the plan file.

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 7-1).

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 7-1.

Table 7-1 WAE Core Configuration Files

Configuration File	Description
`com.cisco.wano.nsps.demand.persist impl.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.c fg`	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 7-2.

Table 7-2 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
```

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.
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

Snapshot Examples

The following sections are **partial** examples that identify the options required or useful when discovering specific features. These examples focus on requirements and anomalies. They do not represent all the possible tasks and CLI options. All examples assume you have defined the environment variables and called other tasks in snapshot.txt, and that you have properly configured the snapshot.inc file for all other tasks.

Collecting Segment Routing LSPs

Step 1 In snapshot.txt, confirm that the following tasks are enabled (uncommented):

- LOGIN_FIND_IGP_DB
- SNMP_FIND_NODES
- SNMP_FIND_INTERFACES
- IMPORT_PCEP_LSPS
- GET_CONFIGS
- PARSE_CONFIGS
- SNMP_FIND_RSVP

Step 2 In snapshot.inc, add or set the following options:

Name	Required Value
<LOGIN_FIND_IGP_DB_CMD_OPT>	
get-segments	true
<SNMP_FIND_RSVP_CMD_OPT>	
keep-pcep-paths	true
get-pcep-paths	false
<PARSE_CONFIGS_CMD_OPT>	
include-object	BASE, RSVP, SR_LSPS

For example:

```
<LOGIN_FIND_IGP_DB_CMD_OPT>
Name    Value
```

```
out-file        $(work_dir)/$(unique).txt
seed-router     $(seed_router)
backup-router   $(backup_router)
igp-protocol    $(igp)
isis-level      $(isis_level)
ospf-area       $(ospf_area)
get-segments    true
session-type    $(session_type)
database-file   $(debug_dir)/$(unique)-$(igp)_db.txt
verbosity       $(cmd_verbosity)
log-file        $(log_dir)/$(unique)-log-login_find_igp_db.log

<SNMP_FIND_RSVP_CMD_OPT>
Name    Value
out-file        $(work_dir)/$(unique).txt
plan-file       $(work_dir)/$(unique).txt
net-recorder    $(net_recorder)
net-record-file $(data_dir)/$(unique)-record-snmp_find_rsvp.txt
verbosity       $(cmd_verbosity)
log-file        $(log_dir)/$(unique)-log-snmp_find_rsvp.log
use-signaled-name       true
get-backup-paths        true
keep-pcep-paths true
get-pcep-paths  false

<PARSE_CONFIGS_CMD_OPT>
Name    Value
igp-protocol    $(igp)
isis-level      $(isis_level)
include-object  BASE,RSVP,SR_LSPS,LAG,SRLG,RSVP,VPN
out-file        $(work_dir)/$(unique).txt
data-dir        $(cfg_dir)
log-level       $(log_level)
log-file        $(log_dir)/$(unique)-log-parse_configs.log
```

Insert Data into External Archive

This example shows how to insert data into an external archive where the information is available for all applications to use.

Step 1 In the snapshot.txt, point the default `archive_dir` to point to the external archive. Best practice is to keep the default.

Example: `archive_dir $(home_dir)/archives`

Step 2 In snapshot.txt, enable the ARCHIVE_INSERT task (uncomment it).

Step 3 In snapshot.inc, use `archive_insert` to insert WAE Live plan files into the external archive during the collection process.

Name	Required Value
<ARCHIVE_INSERT>	
cmd	$(cariden_home)/bin/archive_insert
cmd_opt	ARCHIVE_INSERT_CMD_OPT
<ARCHIVE_INSERT_CMD_OPT>	
plan-file	$(work_dir)/$(unique).pln
archive	$(archive_dir)/$(unique)-archive
time	$(start_time)

Collecting BGP LS

This example provides the workflow for enabling WAE Collector to get BGP LS from the Open SDN Controller (OSC).

Prerequisites:

- BGP LS must be properly configured on the router. For an example of how to do this, refer to the command reference guide for the Cisco IOS XR router, which you can find here: http://www.cisco.com/c/en/us/support/routers/carrier-routing-system/products-command-reference-list.html.
- For BGP LS to be collected, the Automation server must have the wae-osc, wae-core, and wae-db services running. It is also recommended to have wae-messaging service running.

Step 1 Configure OSC to collect BGP LS. For information, see the *BGP LS PCEP User Guide*:

https://wiki.opendaylight.org/view/BGP_LS_PCEP:User_Guide

Step 2 Configure the snapshot.txt file to specify an environment variable for the BGP LS server URL and to turn on the BGL LS Discovery task.

a. Specify the `bgpls_url` environment variable. The default BGP LS server port on which it listens is 8181.

Example: `bgpls_url http://localhost:8181`

b. Uncomment the FIND_BGPLS task in the snapshot.txt file.

Example:

```
<DISCOVERY_TASKS>
#SAM_GETPLAN
#SNMP_FIND_OSPF_DB
#LOGIN_FIND_IGP_DB
FIND_BGPLS
SNMP_FIND_NODES
SNMP_FIND_INTERFACES
```

Step 3 Configure the snapshot.inc file to specify how you want to collect the BGP LS data. The default is to collect OSPF, area 0. This example demonstrates how to collect using both IS-IS levels.

Name	Required Value
<FIND_BGPLS>	
cmd	$(cariden_home)/bin/find_bgpls
cmd_opt	FIND_BGPLS_CP_CMD_OPT
<SNMP_FIND_BGPLS_CMD_OPT>	
url	$(bgpls_url)
out-file	$(work_dir)/$(unique).txt
igp-protocol	isis
isis-level	2
log-file	$(log_dir)/$(unique)-log-find_bgpls.log

Step 4 On the Automation server, start or restart the wae-osc service.

```
service wae-osc start
service wae-osc restart
```

Collecting BGP Peers

Step 1 In snapshot.txt, confirm that the following tasks are enabled (uncommented):

- LOGIN_FIND_IGP_DB (or SNMP_FIND_OSPF_DB)
- SNMP_FIND_NODES
- SNMP_FIND_INTERFACES
- FIND_BGP

For example:

```
<DISCOVERY_TASKS>
Task
#SAM_GETPLAN
SNMP_FIND_OSPF_DB
#LOGIN_FIND_IGP_DB
#FIND_BGPLS
SNMP_FIND_NODES
SNMP_FIND_INTERFACES
#IMPORT_PCEP_LSPS
#GET_CONFIGS
#PARSE_CONFIGS
FIND_BGP
#SNMP_FIND_RSVP
#SNMP_FIND_VPN
```

Collect eBGP Peers by MAC Address

This example shows how to discover and poll eBGP peers by MAC address using manual snapshots. This feature provides more granular traffic collection for networks that establish BGP peering with a large number of ASNs through switch interfaces at public Internet exchange points (IXPs).

Note MAC accounting must be enabled on the routers.

Step 1 In snapshot.inc, use `find-bgp` with the `-get-mac-address` option set to `true`. This enables discovery of eBGP peers by MAC addresses.

Name	Required Value
<FIND_BGP>	
cmd	$(cariden_home)/bin/find_bgp
cmd_opt	FIND_BGP_CMD_OPT
<FIND_BGP_CMD_OPT>	
get-mac-address	true

Step 2 In snapshot.inc, use `snmp_poll` with the `-poll-function` option set to a value that specifies both `interface` and `mac`. This collects interface traffic statistics by MAC addresses.

Name	Required Value
<SNMP_POLL>	
cmd	$(cariden_home)/bin/snmp_poll
cmd_opt	SNMP_POLL_CMD_OPT
<SNMP_POLL_CMD_OPT>	
poll-function	interfaces, mac
polling-interval	interfaces=60, mac=60
number-of-samples	interfaces=1,mac=1

Step 3 In snapshot.txt, ensure that <FIND_BGP> and <SNMP_POLL> are enabled.

Collect Data for WAE Live

For Explore and Analytics components, this example shows how to set up the collection of the statistics that are put into the data store.

Step 1 In snapshot.txt, ensure the following are enabled:

- <SNMP_FIND_NODES>

- Either <SNMP_POLL> or <SNMP_POLL_INTERFACES>, depending on which has its `-perf-data` option set to true in the snapshot.inc file.

Step 2 In snapshot.inc, set the `-perf-data` option to true for `snmp_find_nodes`.

Name	Required Value
<SNMP_FIND_NODES>	
cmd	$(cariden_home)/bin/snmp_find_nodes
cmd_opt	SNMP_FIND_NODES_CMD_OPT
<SNMP_FIND_NODES_CMD_OPT>	
perf-data	true

Step 3 In snapshot.inc, set the `-perf-data` option to true for either `snmp_poll` or `snmp_poll_interfaces`.

	Either...	Or...
	<SNMP_POLL>	**<SNMP_POLL_INTERFACES>**
cmd	$(cariden_home)/bin/snmp_poll	$(cariden_home)/bin/snmp_poll_interfaces
cmd_opt	SNMP_POLL_CMD_OPT	SNMP_POLL_INTERFACES CMD_OPT
	<SNMP_POLL_CMD_OPT>	**<SNMP_POLL_INTERFACES_CMD_OPT>**
perf-data	true	true

Step 4 If analyzing LAGs in WAE Live, set the `snmp_find_interfaces -lag` option to `true`. See Collect LAG Membership and Traffic.

Manually Insert WAE Live Data

This example shows how to insert data directly into the WAE data store and Map archive, rather than storing on a server or in an external archive.

Insert Data into Data Store

Step 1 In snapshot.txt, enable the ML_INSERT task (uncomment it).

Step 2 In snapshot.inc, use <ML_INSERT> to insert plan files into the WAE Live data store during the collection process.

Name	Required Value
<ML_INSERT>	
cmd	$(cariden_home)/bin/ml_insert_plan
cmd_opt	ML_INSERT_CMD_OPT
<ML_INSERT_CMD_OPT>	
plan-file	$(work_dir)/$(unique).pln
time	$(start_time_direct)

Insert Data into Map Archive

This is only applicable if using `ml_insert_plan` and if using the WAE Live Map component. The location specified must be the location of the Map archive directory. This is not the same as the external archive.

Step 1 In the snapshot.txt, create an environment variable that specifies the location of the Map archive.

Example: `map_archive_dir $(home_dir)/data/mldata/archive`

Step 2 In snapshot.txt, add an MAP_ARCHIVE_INSERT task.

Step 3 In snapshot.inc, add <MAP_ARCHIVE_INSERT> to insert WAE Live plan files into the internal Map archive during the collection process.

Name	Required Value
<MAP_ARCHIVE_INSERT>	
cmd	$(cariden_home)/bin/archive_insert
cmd_opt	MAP_ARCHIVE_INSERT_CMD_OPT
<MAP_ARCHIVE_INSERT_CMD_OPT>	
plan-file	$(work_dir)/$(unique).pln
archive	$(map_archive_dir)/archive
time	$(start_time)

Collect LAG Membership and Traffic

Step 1 In snapshot.txt, ensure both <SNMP_FIND_INTERFACES> and <SNMP_POLL> are enabled.

Step 2 In snapshot.inc, use `snmp_find_interfaces` to discover LAG ports with the `-lag true` option. This populates the <Ports> and <PortCircuits> tables. The latter is based on a best-match rule according to ascending port names and numbers.

Use the `-lag-port-match` option to specify how ports are matched in port circuits. Here, the `complete` value is used to tell WAE Collector that if a port is up, match it deterministically based on LACP, and if a port is down, use the `guess` mode to match as many ports as possible.

Name	Required Value
<SNMP_FIND_INTERFACES>	
cmd	$(cariden_home)/bin/snmp_find_interfaces
cmd_opt	SNMP_FIND_INTERFACES_CMD_OPT
<SNMP_FIND_INTERFACES_CMD_OPT>	
lag	true
lag-port-match	complete

Step 3 In snapshot.inc, use `snmp_poll` to poll all LAG and bundle ports for traffic measurements with the `-poll-function ports` option. Ports are polled with the same parameters as interfaces.

Name	Required Value
<SNMP_POLL>	
cmd	$(cariden_home)/bin/snmp_poll
cmd_opt	SNMP_POLL_CMD_OPT
<SNMP_POLL_CMD_OPT>	
poll-function	interfaces, ports
polling-interval	interfaces=60
number-of-samples	interfaces=1

Collect QoS and Traffic

Step 1 In snapshot.txt, ensure both <SNMP_FIND_NODES> and <SNMP_POLL> are enabled.

Step 2 In snapshot.inc, use `snmp_find_nodes` to discover interface queues with the `-read-qos-queues true` option.

Name	Required Value
<SNMP_FIND_NODES>	
cmd	$(cariden_home)/bin/snmp_find_nodes
cmd_opt	SNMP_FIND_NODES_CMD_OPT
<SNMP_FIND_NODES_CMD_OPT>	
read-qos-queues	true

Step 3 In `snapshot.inc`, use `snmp_poll` to poll all interface queues with the `-qos-queues '*'` option.

Name	Required Value
<SNMP_POLL>	
cmd	$(cariden_home)/bin/snmp_poll
cmd_opt	SNMP_POLL_CMD_OPT
<SNMP_POLL_CMD_OPT>	
poll-function	interfaces
polling-interval	interfaces=60
number-of-samples	interfaces=1
qos-queues	‘*’