



Cisco WAE 6.2 Platform Configuration Guide

First Published: June 2015

Last Updated: September 17, 2015

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.

Text Part Number:

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. (1110R)

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.

© 2015 Cisco Systems, Inc. All rights reserved.



WAE Platform Overview	1-1
Modules	1-2
Configuration	1-3
Terminology	1-3
Related Topics	1-4
WAE Core Server	2-1
Using this Chapter	2-1
WAE Core Server	2-1
Processes	2-1
Memory	2-2
Logging	2-2
WAE REST API	2-2
WAE Thrift API	2-3
Deployer Module	3-1
Deployer Configuration	3-1
WAE Collector Overview	4-1
Workflow	4-3
WAE Collector and Archives	4-3
WAE Collector and Applications	4-4
Related Topics	4-5
WAE Collector UI Overview	5-1
Interval Collections and Continuous Polling	5-1
UI Overview	5-2
Workflow	5-2
Management and Troubleshooting	5-3
Common Table Tools	5-3
Related Topics	5-3
Node Discovery	6-1
Node Discovery Fields	6-1

- IGP Discovery 6-1
 - Direct Node Discovery 6-2
 - Node List and Apply 6-2
 - Related Topics 6-2
- Node Access 7-1**
 - Node Access Fields 7-1
 - Optional Fields 7-2
 - Node List and Apply 7-2
 - Related Topics 7-2
- Node Inclusion 8-1**
 - Node Inclusion Fields 8-2
 - Node List and Apply 8-2
 - Related Topics 8-2
- Node List 9-1**
 - Override Rules 9-1
 - Node Inclusion 9-2
 - Node List Table Columns 9-2
 - Status Columns 9-2
 - Property Columns 9-2
 - Node List Options 9-3
 - Edit and Delete 9-3
 - Test and Apply 9-3
 - Related Topics 9-3
- What to Collect 10-1**
 - What to Collect Fields 10-1
 - Basic 10-1
 - BGP/VPN 10-2
 - Advanced Configurations 10-3
 - Related Topics 10-3
- Schedule 11-1**
 - Schedule Fields 11-1
 - Related Topics 11-2
- Status 12-1**
 - Last Snapshot Status 12-1

Collection Metrics	12-1
Status Summary	12-1
Related Topics	12-2
Logs	13-1
Related Topics	13-1
WAE Network Interface Server	14-1
Continuously Polling	14-1
Workflow	14-2
WAE NI server Fields	14-3
Server Access	14-3
Server Configuration	14-3
Related Topics	14-3
Configuration	15-1
Configuration Fields	15-1
Related Topics	15-2
Advanced WAE Collector Configurations	16-1
Terminology	16-2
Related Topics	16-2
Snapshot Files	17-1
Snapshot Configuration Files	17-1
snapshot.txt	17-2
Environment Variables	17-2
snapshot.txt Tasks	17-3
snapshot.inc	17-4
Launch and Validate snapshot	17-5
Related Topics	17-6
Augmented Collection	18-1
Parameters	18-1
Workflow	18-2
Configure the Server	18-2
Configure Credentials	18-3
Pre-Snapshot Configuration	18-3
Configure Augmented Snapshot Files	18-3
Initialize Archive, Create Template, Run Collections	18-5

Using Collections 18-6

Related Topics 18-6

Manual Collection 19-1

Workflow 19-1

Pre-Snapshot Configuration 19-1

Modify Snapshot Files 19-2

Initialize Archive, Create Template, Run Collections 19-4

Using Collections 19-5

Snapshot Examples 19-5

 Insert Data into External Archive 19-6

 Collect BGP LS 19-6

 Collect Data for WAE Live 19-7

 Manually Insert WAE Live Data 19-8

 Collect LAG Membership and Traffic 19-9

 Collect eBGP Peers by MAC Address 19-10

 Collect QoS and Traffic 19-11

Related Topics 19-12

Continuous Polling and PCEP LSP Collection 20-1

Workflow 20-1

Set Up WAE NI Server 20-2

 Configure Continuous Parameters 20-2

 Configure Authentication and Start Server 20-4

Pre-Snapshot Configuration 20-4

Create Snapshot to Push Plan Files 20-5

 Configure Push Credentials 20-5

 Modify Push snapshot.txt 20-5

 Modify Push snapshot.inc 20-6

Create Snapshot to Get Plan Files 20-8

 Configure Get Credentials 20-8

 Modify Get snapshot.txt 20-8

 Modify Get snapshot.inc 20-10

 Initialize Archive and Create Template 20-11

Run Collections 20-11

Using Collections 20-12

Related Topics 20-12

Multi-Network Collection	21-1
Prerequisites	21-1
Pre-Snapshot Configuration	21-2
Using External Archives	21-3
Inserting Plan Files Directly	21-5
Related Topics	21-7
Flow Collection	22-1
Workflow	22-2
Configuration	22-3
Manage Flow Collection	22-4
<NodeFlowConfigs> Table	22-4
flow_manage	22-5
Get Traffic Matrices into a Plan File	22-5
Demands	22-5
flow_get	22-6
Collect Flows	22-7
Snapshot Integration	22-7
Related Topics	22-8
Hardware Inventory Collection	23-1
Configure Inventory Collection	23-1
Pre-Snapshot Configuration	23-2
Configure Snapshot Files	23-2
Test and Run Collections	23-6
Use Inventory in WAE Live	23-7
Edit Default Inventory Files	23-7
Collected Hardware	23-7
Tables for Processing Inventory	23-8
Related Topics	23-12
Offline Discovery	24-1
Import Databases	24-1
Import Router Configuration Files	24-1
Import IGP Database	24-3
get_show	24-6
Import Traffic from RRD Tools	24-6
Cricket	24-6
Cacti	24-6

MRTG 24-7

Related Topics 24-7

SAM Integration 25-1

SAM Discovery 25-1

SDPs 25-2

Configure SAM for Use with WAE Collector 25-3

Configure SAM Group and User Account with OSS Permission 25-3

Configure SAM to Collect and Store Performance Statistics 25-5

Configure SAM to Collect and Store Accounting Statistics 25-6

Apply Accounting Policies 25-9

Verify Accounting Statistics Collection 25-11

SAM Integration 25-17

sam_getplan 25-18

Snapshot Integration 25-19

Related Topics 25-19

Network Authentication 26-1

Online Discovery Authentication 26-1

Create an Authentication File 26-1

Tables in the Authentication File 26-3

Add Router-Specific Authentication Information 26-4

View Authentication Information 26-5

Test the Authentication File 26-5

Related Topics 26-5

Network Access File 27-1

File Format 27-1

Global Settings 27-2

Per Router Settings 27-3

Test the Network Access File 27-4

Tool Access Parameters 27-4

Related Topics 27-5

Manage Archives 28-1

Create or Update an Archive 28-1

Update Summary of Time-Sequence Plot Data 28-2

Insert or Extract Files from an Archive 28-2

Insert Files 28-2

Extract and Delete Files	28-3
Manage Archives for WAE Design Archive	28-3
Make Batch Changes to Archive Files	28-3
Related Topics	28-4
WAE Design GUI and Remote WAE Core Server	29-1



CHAPTER 1

WAE Platform 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.

The WAE platform creates and maintains 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. 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.

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.

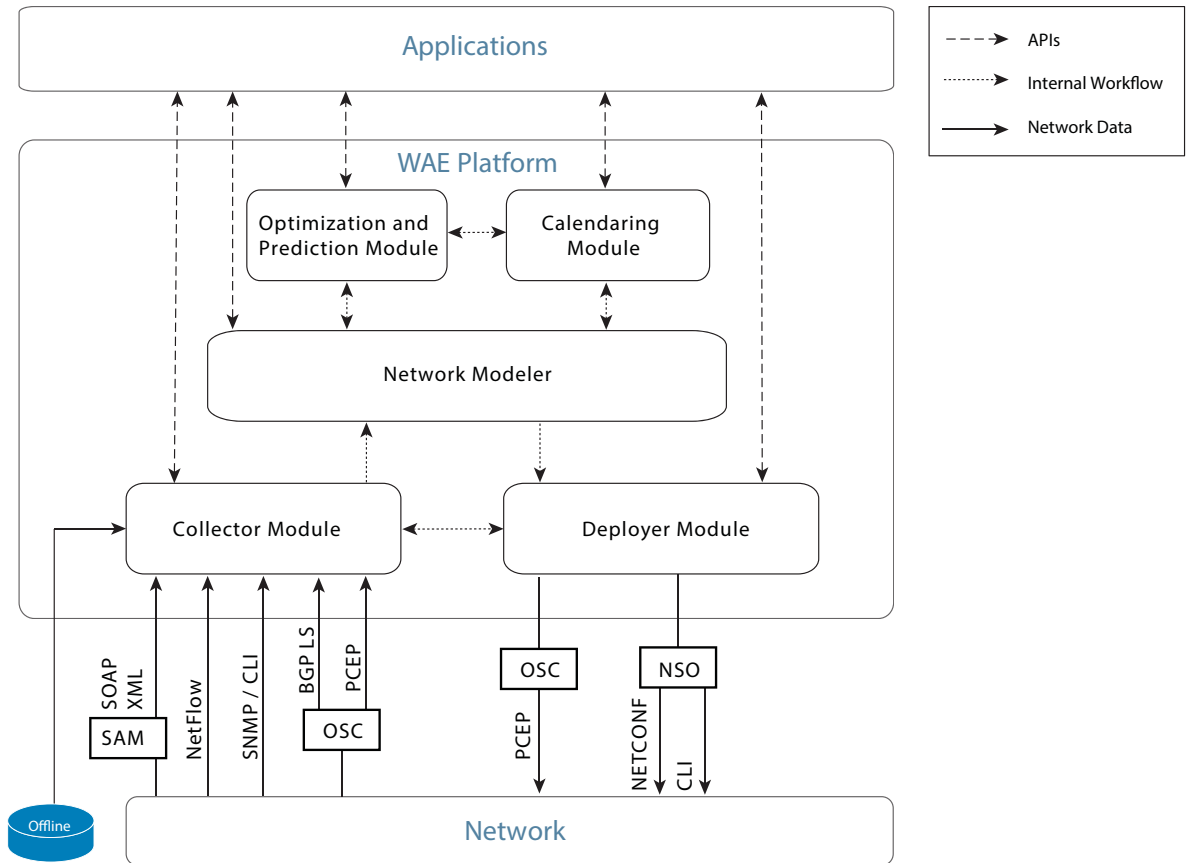
- **Optimization and Prediction Module**—Enables APIs to execute global and tactical optimizations, model traffic-engineered LSPs, and perform failure analysis. These simulation tools enable you to model and predict how the network will react to a specified list of changes, as well as suggest additional 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



353688

Configuration

This guide describes the following configurations.

- [WAE Core Server](#)—Describes how to configure WAE Core server resources for process threads, memory usage and logging, as well as how to configure the REST and Thrift APIs.
- [Deployer Module](#)—Describes how to configure the deployment of PCEP LSPs.
- [WAE Collector Overview](#)—Describes the differences between the collection methods and helps you identify which one is best for you. Note that the WAE Collector configuration chapters are extensive. This overview chapter guides you to further exploring these configurations.

Terminology

This guide uses the following terms.

- `$CARIDEN_ROOT` and `$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`.

Related Topics

- *WAE Live Configuration Guide*
- *WAE Design Archive User and Administration Guide*
- *WAE Design Plan Table Schema and CLI Reference*



WAE Core Server

The WAE Core server has numerous configuration files. This chapter describes the most likely configuration changes within these files and does not describe all the available configuration files or options.

- [WAE Core Server](#)
 - Processes
 - Memory
 - Logging
- [WAE REST API](#)
- [WAE Thrift API](#)

Using this Chapter

- This chapter references `$WAE_ROOT`, which is the installation location. The default `$WAE_ROOT` is `/opt/cariden`.
- 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. The instructions in this chapter assume this knowledge and do not repeat it at every step.



Note

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

WAE Core Server

Processes

Configuration file: `$WAE_ROOT/software/wae-core/etc/com.cisco.wano.nsp.engine.cfg`

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=<#>
```

Memory

Configuration file: `$WAE_ROOT/software/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_ROOT/software/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
<code>log4j.logger.com.cisco=<log_level></code> Example: <code>log4j.logger.com.cisco=TRACE</code>	DEBUG	The type of log level to use can be ERROR, WARN, INFO, DEBUG, or TRACE.
<code>log4j.appender.out.maxFileSize=<whole_number>[MB GB]</code>	10MB	Maximum permissible log file size.
<code>log4j.appender.out.maxBackupIndex=<whole_number></code>	10	Maximum permissible number of backup log files.

WAE REST API

Configuration file: `$WAE_ROOT/software/wae-core/etc/com.cisco.wano.nsps.nbrs.cfg`

To manage the behavior of the REST northbound API, set these properties. Note that 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.

To Change	Change This Property Value	For HTTP	For HTTPS
Protocol	nbrsProtocol	http	https
REST service port	nbrsPort	The default is 7777. You can keep or change this value.	
SSL port	jettyEngineFactoryPort	The default is 7776. This port must be different than the nbrsPort value, and it must not be in use elsewhere.	The nbrsPort port and jettyEngineFactoryPort must be the same value. The easiest way to do this is to uncomment the one that uses the nbrsPort variable and comment out the one setting a specific value for the SSL port. jettyEngineFactoryPort=\${nbrsPort} #jettyEngineFactoryPort=7776

- If receiving timeout errors, increase the timeout value.
nbQSendOptions=?requestTimeout=<# of milliseconds>

WAE Thrift API

Configuration file: \$WAE_ROOT/software/wae-core/etc/com.cisco.wano.nsp.thrift.cfg

To adjust the settings for the Thrift northbound API, use these 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>



Deployer Module

The Deployer Module communicates with PCEs in the network for changing, adding, or deleting LSP path configurations.

This chapter references `$WAE_ROOT`, which is the installation location. The default `$WAE_ROOT` is `/opt/cariden`.

You can also deploy LSPs using the WAE Design GUI. For information, see the [WAE Design GUI and Remote WAE Core Server](#) chapter.



Note

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

Deployer Configuration

Configuration File: `$WAE_ROOT/software/wae-core/etc/com.cisco.wano.nsps.deployer.pcep.cfg`

Identify how to handle deployment failures by setting this `deployerFailurePolicy` property. There are two options.

- **BEST_EFFORT**—Once the failure occurs, continue deploying as much as possible. To determine the deployment state, use the following API.
`/wae/network/deployer/job/jobState`
- **STOP_ON_FAILURE**—Stop the deployment immediately upon failure, and deploy nothing.

Example: `deployerFailurePolicy=STOP_ON_FAILURE`

Tell the Deployer which proxy to use for PCEP configurations by setting the `pcepDeployerProxy` property. There are two options.

- **testPcepDeployerProxy** (default)—Invoke the PCEP Deployer, but do not communicate with the OSC controller.
- **odlPcepDeployerProxy**—Invoke the PCEP Deployer using this proxy. You must set this parameter with this option if using OSC to discover PCEP tunnels.

Example: `pcepDeployerProxy=odlPcepDeployerProxy`

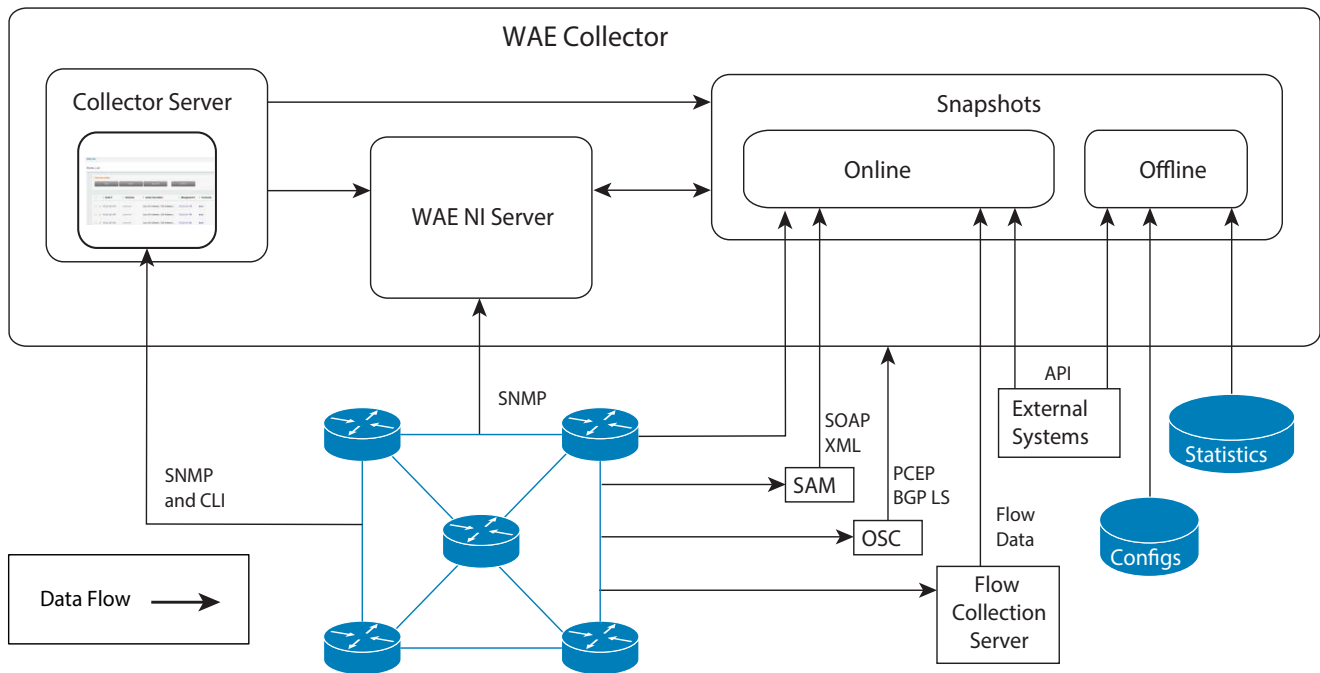


CHAPTER 4

WAE Collector Overview

WAE Collector discovers topology, traffic, and operational information for multi-vendor networks through a combination of online and offline tools. Three methods of discovery are supported (Figure 4-1). The simplest configuration is through the Collector server, which is configured entirely through the WAE Collector UI; from here you can connect to the WAE Network Interface (NI) server to continuously poll traffic statistics, as well as to continuously collect PCEP LSPs. The augmented method augments the UI output with manual CLI collection steps. There is also an entirely manual CLI method that relies solely on configuration files.

Figure 4-1 WAE Collector Workflow



To determine which approach to use, consider what it is you are trying to discover and what the application needs are (Table 4-1¹). Regardless of the method, the basic unit of data storage that is transferred between WAE Collector and the applications is a *plan file*.

1. This table does not include the advanced configuration options available in the WAE Collector UI. Additionally, all collections are dependent on licenses and what you have configured for collection.

- **Collector Server**—The Collector server method of collection produces a plan file that can be further processed using augmented snapshots².
- **Augmented**—The augmented method is primarily for preparing the plan file for use in WAE Design and WAE Design Archive. This method retrieves a plan file from the Collector server or WAE NI server, optionally 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.
- **Manual**—The manual method is used for advanced configurations that are not supported by the Collector server. It uses snapshot configuration files for collection. Examples include collection directly from config files, multi-networking collection, and collection from Alcatel-Lucent's 5620 Service Aware Manager (SAM) server.

Table 4-1 Collection by Configuration Method

	Collector Server ¹	Augmented Snapshot	Manual Snapshot
Uses SNMPv2c authentication	x	x	x
Uses SNMPv3 authentication			x
Directly discovers nodes using system IPv4 addresses	x		
Collects OSPF and IS-IS IPv4 topologies	x	x	x
Collects OSPF and IS-IS IPv6 topologies			x
Collects BGP LS topologies			x
Collects node properties	x	x	x
Collects interface properties, including TE extensions	x	x	x
Collects interface queues	x	x	x
Collects SRLGs		x	x
Discovers BGP peering	x	x	x
Continuously polls traffic statistics (requires the WAE NI server)	x	x	x
Continuously collects Collects PCEP LSPs ² (requires the WAE NI server)	x	x	x
Collects basic RSVP LSP properties	x	x	x
Collects RSVP LSPs with multiple paths or named paths (EROs)		x	x
Collects LAG ³ ports	x	x	x
Collects Multicast		x	x

2. The term *snapshot* means “captured view of the network,” “collection,” or “collection process.” For example, you can take a snapshot (captured view of the network) using snapshot (collection) configuration files. Each snapshot (collection process) runs for a period of time.

	Collector Server ¹	Augmented Snapshot	Manual Snapshot
Collects VPNs	x (Layer 3 only)	x	x
Collects LDPs		x	x
Collects flow traffic		x	x
Collects topology from config files			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 WAE Collector UI. Additionally, all collections are dependent on licenses and what you have configured for collection.

2. Only PCEP LSPs that are deployed to the network through WAE Core APIs are collected.

3. 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*.

Workflow

Figure 4-2 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 *WAE Network Visualization Guide*.

WAE Collector and Archives

The first step is for WAE Collector to discover the network and create a plan file that represents your network. These plan files capture all relevant information about a network at a given time, and can include topology, configuration, traffic, routing, and related information.

- 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 and/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 on the WAE Core server.

**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 and/or for the continuous collection of PCEP LSPs. From a snapshot, you can also pull a plan file from the WAE NI server.

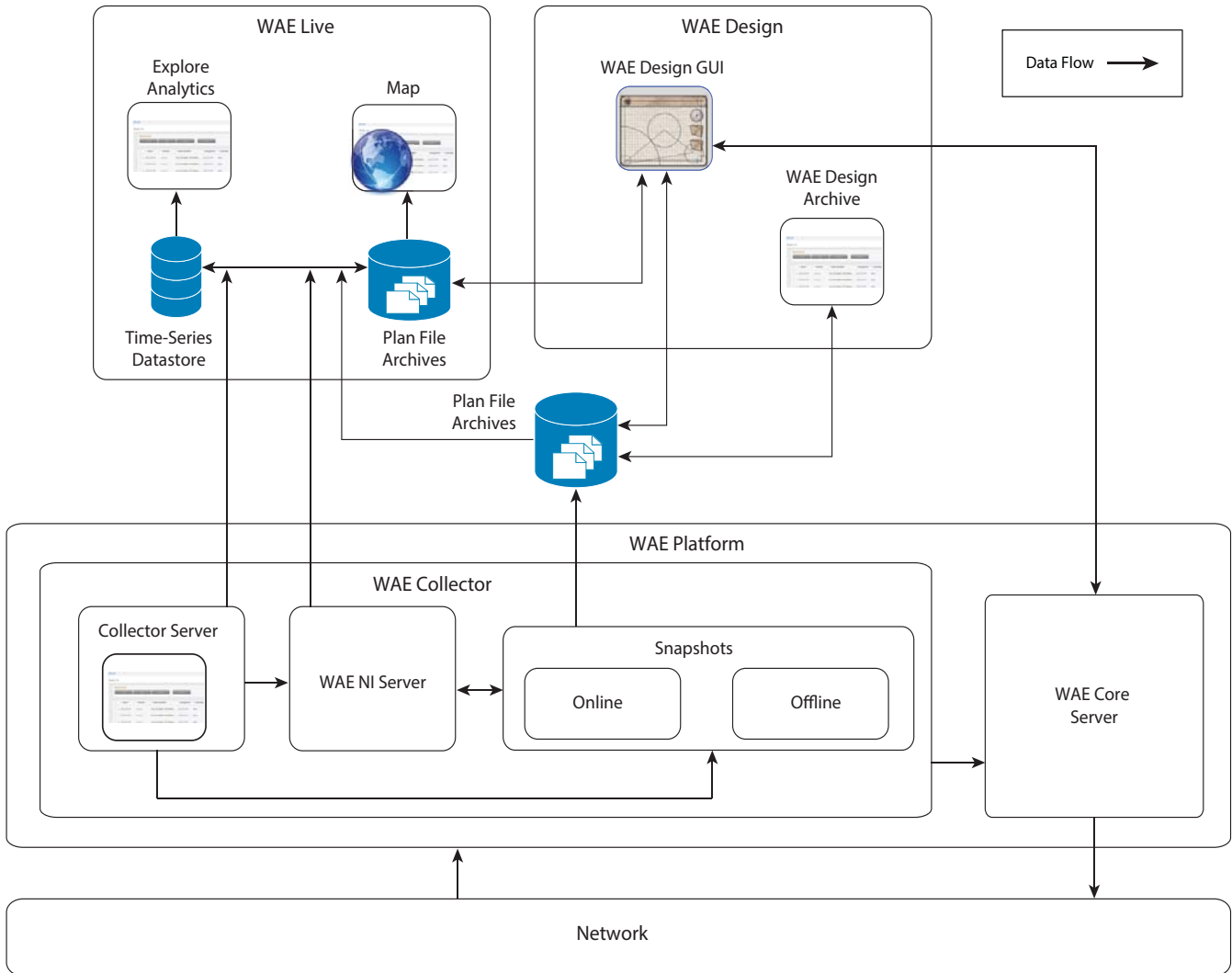
WAE Collector and Applications

All WAE applications use plan files produced by WAE Collector.

- 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.

The WAE Design GUI can also access plan files residing in the Network Modeler Module on the WAE Core server, as well as save to the plan file back to the WAE Core server. You can additionally deploy the plan file to the network directly from WAE Design. (A copy of it is also saved to the working plan file area in the Network Modeler Module.)

Figure 4-2 Data Flow after Collection



353689

Related Topics

- [WAE Collector UI Overview](#)
- [Advanced WAE Collector Configurations](#)
- [WAE Network Visualization Guide](#)
- [WAE Design Integration and Development Guide](#)
- [WAE Design Plan Table Schema and CLI Reference](#)



WAE Collector UI Overview

The WAE Collector UI enables you to configure the collection of basic network data. 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 Network Interface (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 PCEP LSPs to the WAE NI server.

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

- If continuous polling is not running and if PCEPs 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 PCEPs 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.

Interval Collections and Continuous Polling

Through the WAE Collector UI, you have the option to run collection based on intervals alone 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 the 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 on the [WAE Network Interface Server](#) page. 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.

UI Overview

The following table describes the basic components available through the WAE Collector UI.

Menu	Description
Setup	Configure global rules for defining the set of nodes from which to collect network data.
Node List	View discoverable nodes and configure per-node access rules that override the global rules.
Collection	<ul style="list-style-type: none"> Configure which properties and traffic to collect. Start, stop, and troubleshoot collections. Schedule collection frequency. View status of most recent collection and log files for all collections since starting the Collector server. These are applicable only to the Collector server.
Settings	<ul style="list-style-type: none"> Save and load configuration files, and reset UI settings to their defaults. Connect to the WAE NI server and enable continuous collection of PCEP LSPs and/or enable continuous polling. This page also enables you to configure time intervals for averaging the continuously polled traffic statistics.

Workflow

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

If another person is accessing the WAE Collector UI when you log in, you will receive a message as such. Note that any changes you make will affect the other person's configuration, and vice versa.

Step	Description	For Information
1	If collecting PCEP LSPs or if continuously polling traffic statistics, configure the authentication for the WAE NI server and start it.	WAE Network Interface Server
2	Configure the list of nodes to collect and their authentication details. <ol style="list-style-type: none"> Setup->Node Discovery: Configure the IGP protocol for collecting the nodes using the seed router, configure the collection using system IPv4 addresses, or both. Setup->Node Access: Configure global access rules and global inclusion rules that determine from which nodes to collect. Setup->Node Inclusion: If needed, using regular expressions, further configure which nodes to include or exclude. 	Node Discovery Node Access Node Inclusion
3	Node List: Verify the node list, and if needed, individually configure per-node access rules that override the global ones.	Node List

Step	Description	For Information
4	Collection->What to Collect: Configure which objects and traffic to collect, as well as the counter polling period.	What to Collect
5	Settings->WAE NI Server: If collecting PCEP LSPs or if continuously polling traffic statistics, connect to the WAE NI server. Settings->WAE NI Server: If continuously polling, enable the statistics collection and configure the time interval for averaging the polled traffic.	WAE Network Interface Server
6	Collection->Schedule: Schedule how frequently you want to collect the network data.	Schedule
7	Collection->Schedule: Start the collection process.	Schedule

Management and Troubleshooting

Throughout the collection process, 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.

On the Schedule page, 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.

Common Table Tools

- Filter—Use this feature to narrow the results in the table. This enables you to find specific information faster or fine-tune the list for troubleshooting specific nodes.
- Sorting—Use the column headings to sort the tables by the relevant column. This is useful, for example, to find nodes that are not accessible via SNMP or login.
- Gear icon—Use this icon to specify the number of rows to show in the table.

Related Topics

- Global collection rules
 - [Node Discovery](#)

- [Node Access](#)
 - [Node Inclusion](#)
- Per-node rules that override the global rules
 - [Node List](#) page enables you to identify which nodes are accessible and to be included in the collection, and enables you to individually modify access parameters to selected nodes.
- [What to Collect](#) page enables you to define which objects and traffic to collect, and set the counter polling periods.
- [Schedule](#) page enables you to schedule, start, and stop the collection process.
- [WAE Network Interface Server](#) page enables you to connect to the WAE NI server, configure the time window for averaging the polled traffic, enable continuous collection of PCEP LSPs, and enable continuous polling.
- [Node List](#) page shows node and node-independent issues for the last collection process, as well as snapshot runtime statistics and license violations.
- [Logs](#) page shows details of errors and warnings for all collections.
- [Configuration](#) page enables you to save and load configuration files, and reset UI settings to their defaults.
- [Augmented Collection](#)



Node Discovery

Access: WAE Collector->Setup->Node Discovery

Rule Type: Global seed router and backup router access rules

Node Discovery enables you to discover the nodes in the network by reading the IGP database of a seed router or by specifying a list of system IP addresses. Note that you can combine these methods to populate the node list.

- **IGP Discovery**—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.

Access options include SSH and Telnet, and IGP options include OSPF or IS-IS. Note that 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.

- **Direct Node Discovery**—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.

You can edit this page at any time. Doing so changes the nodes available for collection.

The rules used to collect from nodes can be overwritten by on a per-node basis. If this is the first time you are setting up the node list, continue to configuring [Node Access](#) and [Node Inclusion](#). Thereafter, if you continue to see a need to create per-node overrides, use the [Node List](#) page.

Node Discovery Fields

IGP Discovery

- Discover using IGP Database—Toggle for using 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.
- Initial Authentication—Select whether to log into the seed router or use SNMPv2 to access it. If discovering IS-IS, you must select Login.
- Username—Username for login access to the seed router.

- Password—Password for login access to the seed router
- 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.
- Select IGP—You must select which IGP database to use.
 - OSPFv2—Select if collecting an OSPF database. Collect from a single OSPF area by selecting Specify and entering an area ID. The seed router must belong to the area specified. Collect from all areas by selecting All. 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.
 - IS-IS—Select if collecting an IS-IS 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. Note that node names are available if using IS-IS, and that you must log into the seed router to discover IS-IS.

Optional Fields

- Enable Password—The static password that controls access to the privileged EXEC (enable) mode of the Cisco router.
- Use Backup Seed Router—Toggle that identifies 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—Toggle for using IP addresses to discover nodes. Enter one or more IPv4 addresses separated by commas. You cannot specify a subnet range.

Node List and Apply

For nodes discovered using the IGP Discovery option, selecting Apply first checks if the seed router is reachable, and then uses the login or SNMPv2c information to retrieve the seed router’s IGP database to obtain a list of nodes. If the seed router is not reachable or if you are not satisfied with the nodes being collected, verify the management IP address and other credentials and consider changing the Login session type.

For nodes discovered using the Direct Node Discovery option, selecting Apply uses the specified IP addresses to obtain a list of nodes. If you are not satisfied with the nodes being collected, verify that you have the system IP addresses correctly entered and that the list of IP addresses is complete and accurate.

Related Topics

- Global collection rules

- [Node Access](#)
 - [Node Inclusion](#)
- Per-node rules that override the global rules
 - [Node List](#)
- [What to Collect](#)
- [WAE Network Interface Server](#)



Node Access

Access: WAE Collector->Setup->Node Access

Rule Type: Global node access rules

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. Note that 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 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 configuring [Node Inclusion](#). Thereafter, if you continue to see a need to create per-node overrides, use the page.

Node Access Fields

- Management IP—You must select a method of determining the management IP for nodes.
 - Same as node IP—Select if the node management IP address is the same as the node IP address.
 - Replace node IP prefix with—Use 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: In this example, the node IPs are in the range 5.6.7.8/24, and the management IPs 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 the following.

```
5.6.7.8/24 > 5.6.77.8/24
```

- Default community—SNMPv2c community string that acts as a password. It is used to authenticate messages sent between the node and the seed router.
- Master password—Set a password that enables you to de-encrypt the authentication file. This must contain a lowercase and an uppercase character, a special character, and a number.

Optional Fields

- SNMP access options—Enables you to use either the SNMP community string from the Node Discovery page (if one was set) or set one specifically to reach the nodes.
- Login access options—Provides several options for logging in to routers. 1) Use login information from the Node Discovery page, 2) specify a different username and password when logging in to routers, or 3) disable the login process.



Note

If discovering multi-hop BGP or if you added login tasks using the Advanced Configurations tab on the What to Collect page, you must use a login access. If a default login is not possible, then configure the login access on a per-node basis from the [Node List](#) page.

The following shows the necessary information for logging in using different credentials than the Node Discovery page.

- Username—Username for login access to the nodes.
- Password—Password for login access to the nodes.
- Enable password—The static password that controls access to the privileged EXEC (enable) mode of the Cisco router.

Node List and Apply

Apply populates or updates the Node List table, identifying whether the nodes are reachable via SNMP and if applicable, then login. If you are not satisfied with the nodes being collected, verify the management IP address and other credentials, and consider changing the login session type. If most of them are reachable, then use the per-node override rules (edited from the [Node List](#) page). If most of them are not reachable, verify and change the credentials as needed.

Related Topics

- Global collection rules
 - [Node Discovery](#)
 - [Node Inclusion](#)
- Per-node rules that override the global rules
 - [Node List](#)
- [What to Collect](#)
- [WAE Network Interface Server](#)



Node Inclusion

Access: WAE Collector->Setup->Node Inclusion

Rule Type: Global node inclusion rules

The Node Inclusion page enables you to set global rules for including and excluding nodes 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.*l.+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.

When the configuration is tested or applied, whether a node is included or excluded by rule is indicated by color-coded symbols in the Include column of the [Node List](#) table.

You can edit this page at any time. Doing so changes global rules for whether nodes are included or excluded from being collected.

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.

Node Inclusion Fields

- **Include**—In the collection process, include only nodes that meet the hostname criteria defined by the regular expression.
- **Exclude**—In the collection process, exclude only nodes that meet the hostname criteria defined by the regular expression. If a node is both excluded and included by rules, the exclusion rule takes precedence.

Node List and Apply

Apply updates the Include column in the Nodes List table, identifying whether the nodes are included or excluded based on global rules.

Related Topics

- Global collection rules
 - [Node Discovery](#)
 - [Node Access](#)
- Per-node rules that override the global rules
 - [Node List](#)
- [What to Collect](#)
- [WAE Network Interface Server](#)



Node List

Access: WAE Collector->Node List

Rule Type: Provides ability to create override rules for specific nodes to which the global rules do not apply.

The Node List page displays a table of 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 table 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.

Override Rules

When new nodes appear, WAE Collector tries the global community string in combination with the global management IP. If SNMP access fails, you can get information for these failures on the [Status](#) page. 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, click Edit, and then set these in the Specify option of the Edit field. Alternatively, you can choose to exclude them. See the [Node Inclusion](#) section.

If you do not know this information, use the scan feature available using the Discover option of the Edit field. Here, specify a subnet to search for the management IP and enter one or more SNMP communities. WAE Collector then scans the entire subnet using the entered communities strings in sequence. WAE Collector 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.

Node Inclusion

Nodes are included in the collection or not based on two criteria. 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 this Node List page and on the Status page.

Node List Table Columns

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

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

Property Columns

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.

- 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

These fields are derived as a result of queries to the nodes, and thus are not color-coded.

- 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

Node List Options

Edit and Delete

Edit—The Edit node option enables you to set explicit collection parameters for one or more selected nodes or to exclude selected nodes. Whatever rules you set become immediately applicable to the next collection process once you click Apply. All the global rules (except for those set on the Node Discovery page) are available to change on a per-node basis.

- Exclude from collection—Do not include the selected nodes in the collection.
- Edit
 - 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.

Delete—WAE Collector never dynamically removes nodes from the Node List table, 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 table, you must manually delete it from the list.

Test and Apply

- 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.
- Apply—Apply the configuration, which updates the Node List table.

Related Topics

- Global collection rules
 - [Node Discovery](#)
 - [Node Access](#)
 - [Node Inclusion](#)

- [What to Collect](#)
- [Schedule](#)
- [WAE Network Interface Server](#)
- [Status](#)
- [Logs](#)
- [Configuration](#)



What to Collect

Access: WAE Collector->Collection->What to Collect

After you have verified the node list from which you are working, 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 the applications.

- | | |
|--|---|
| <ul style="list-style-type: none">• Basic properties and traffic<ul style="list-style-type: none">– Nodes– Interfaces– Interfaces queues– RSVP TE LSPs– PCEP LSPs (requires the WAE NI server) | <ul style="list-style-type: none">• BGP connectivity• Layer 3 VPNs and traffic |
|--|---|



Note

Note that all traffic measurements are in Mbps. To poll the traffic continuously, configure the WAE Network Interface (NI) server. For information, see the [WAE Network Interface Server](#) chapter.

Node names collected from the network often have long suffixes that are the same for all nodes. This page enables you to remove these unwanted 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 remove inactive interfaces and circuits from the plan file, thus keeping plan files up to date.

What to Collect Fields

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

Basic

- Interfaces—Collectively identify each interface. For example, an interface's properties could include its interface name, capacity, IGP metric, and TE metric.

- Include queues—The list of interface queues configured on the router, together with per-queue traffic measurements.
- Traffic—Incoming and outgoing traffic on an interface in Mbps.
- Counter polling period—The intervals (in seconds) between successive traffic counter polls.
- LSPs—Collectively identify each LSP. For example, an LSP's properties could include its destination, setup bandwidth, and the actual path of the LSP.
 - Non-PCEP LSPs—LSPs that are discovered using SNMP.
 - PCEP LSPs—LSPs that are managed by WAE. LSPs that are managed by WAE (PCE) 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.)

The WAE NI server must be on and connected on the [WAE Network Interface Server](#) page.

- 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.
- 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.

- 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.
- Select Nodes—Specify which nodes have delayed counter updates. This is specified using a regular expression match on an LSP property.
- Select from—The LSP property on which the regular expression is applied.
- Regexp match—Enter a regular expression to match nodes with delayed counter update.
- Clear nodes—Clears selection of nodes with delayed counters, and clears all selections made within the LSP section.
- 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 following removes 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

Use this tab to configure discovery of eBGP peers and neighboring external AS's, and to discover VPNs and their traffic.

If discovering multi-hop BGP, you must enable login through the Login Access option on the [Node Access](#) page. 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.

- BGP Peer Protocol—Select to discover eBGP peers and neighboring external AS's. Options include searching for BGP peers based on IPv4 addresses, IPv6 addresses, or both.
- Minimum IPv4 prefix length—Minimum prefix length to perform an IPv4 subnet match from 0 to 32.
- Minimum IPv6 prefix length—Minimum prefix length to perform an IPv6 subnet match from 0 to 128.
- Multi-hop discovery by login—Log into the routers to discover the hops between them. This login must be specified on the [Node Access](#) page.
- VPN—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.

Advanced Configurations

Advanced Config—This enables you to add options to existing commands, and to add new commands. Newly added commands must be only for collection purposes. This feature is only for advanced users since modifying the configuration can break the collection process. 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.



Note

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.

Related Topics

- Global collection rules
 - [Node Discovery](#)
 - [Node Access](#)
 - [Node Inclusion](#)
- Per-node rules that override the global rules
 - [Node List](#)
- [Schedule](#)
- [WAE Network Interface Server](#)



Schedule

Access: WAE Collector->Collection->Schedule

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.

Schedule Fields

- **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 instances from overlapping. It also removes collection instances that do not finish because of an error.
- **Collect verbose diagnostics**—Toggle 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.

Related Topics

- [Node List](#)
- [Status](#)
- [Logs](#)



Status

Access: WAE Collector->Collection->Status



Note

The information on this page pertains only to the local Collector server. 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.

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](#) table 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.

Related Topics

- Global collection rules
 - [Node Discovery](#)
 - [Node Access](#)
 - [Node Inclusion](#)
- Per-node rules that override the global rules
 - [Node List](#)
- [What to Collect](#)
- [Schedule](#)
- [Logs](#)



Logs

Access: WAE Collector->Collection->Logs



Note

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.

The Logs page lists all errors and warnings since the Collector server was last started. Therefore, it is a super-set of the information that is listed on the Status page, which is relevant only for the last collection.

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

Related Topics

- Global collection rules
 - [Node Discovery](#)
 - [Node Access](#)
 - [Node Inclusion](#)
- Per-node rules that override the global rules
 - [Node List](#)
- [What to Collect](#)
- [Schedule](#)
- [Status](#)



WAE Network Interface Server

Access: WAE Collector->Settings->WAE Network Interface Server

The WAE Network Interface (NI) server 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 PCEP LSPs that are managed by WAE. LSPs that are managed by WAE (PCE) 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.)

Continuously Polling

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

- The amount of time for each polling interval is set using the Counter Polling Period field on the [What to Collect](#) page.
- The continuous polling window is set on the WAE NI server page. Statistics are averaged across each time period specified in the Default Time Window field. This period starts at the time the plan file is generated and goes backwards for the specified number of minutes. These statistics are added to the plan file.
- If there is any period of time where statistics are not counted, further attempts to collect these statistics can be extended to the percentage of time identified in the Max Expansion field. This field is a percentage of the Default Time Window field.
- When configured through the WAE Collector UI, the WAE NI server generates a plan file every five minutes by default.

Example:

Counter Polling Period = 120 seconds

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 counters are missed, the process of averaging the counters is extended up to 2 minutes (25% of 8 minutes is 2 minutes).

**Note**

For information on changing the WAE NI server username and password, contact your support representative.

Workflow

This workflow is for continuously collecting PCEP LSPs and for continuously polling. This assumes that you have properly set up the node list as you would for any discovery through Setup pages, and thereafter fine-tuned this node list as needed.

Step 1 Configure the authentication for the WAE NI server, and start it.

- a. Enter the username and password for the WAE NI server.

default username: admin

default password: cariden

If the password has changed and you do not know it, contact your administrator or support representative.

- b. If it is not running, start the WAE NI server by entering the following command.

```
service wae-ni start
```

Step 2 On the [What to Collect](#) page, follow these steps.

- If continuously collecting PCEP LSPs, select the PCEP LSPs option.
- If continuously polling traffic, select which objects to collect. For each object, select the Traffic option and enter the polling interval in the Counter Polling Period field.

**Note**

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

Step 3 Connect to the WAE NI server. See the [Server Access](#) section.

- a. Configure the server location and authentication, and click the applicable Apply button.
- b. 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 (step 1a), started the server (step 1b), and correctly entered server location and authentication information (step 3a).

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

Step 4 For continuously polling, follow these steps.

- Select the Continuous Traffic Statistic Collection check box.
- For continuously polling, configure the window for averaging the statistics. See the [Server Configuration](#) section.

Step 5 From the [Schedule](#) page, schedule and start the collection.

WAE NI server Fields

Server Access

You must click the Apply button that is applicable to these fields to set them.

- 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 and 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.

Server Configuration

You must click the Apply button that is applicable to these fields to set them.

- Continuous Traffic Statistic Collection—A toggle that identifies whether you can continuously collect PCEP LSPs or continuously poll when discovering the network and collecting data. In either case, this toggle must be selected. Even if all other fields on this page are set (applied) and even if the WAE NI server is running and connected, PCEP LSPs cannot be collected and traffic cannot be continuously polled unless this option is selected.
- 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.

Related Topics

- Global collection rules
 - [Node Discovery](#)

- [Node Access](#)
 - [Node Inclusion](#)
- Per-node rules that override the global rules
 - [Node List](#)
- [What to Collect](#)
- [Schedule](#)



Configuration

Access: WAE Collector->Settings->Configuration

A configuration file contains the discovered objects and properties, as well as the configurations used to discover them. The Settings page supports the following related features.

- **Load**—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**—Saves the current configuration file to `<install_directory>/etc/collector/server/configs`. The default installation directory is `/opt/cariden`.
- **Reset**—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.

Configuration Fields

- **Configuration options**—Select whether to save, load, or reset.
- **Path to configuration (Load)**—Click the folder, and then select from the list of saved configuration files.

After loading a configuration, you must re-apply the configurations on the Node Discovery and Node Access pages.

- **Name of the configuration file (Save)**—Enter the name by which you want to save the current configuration file. File names by the same name in `<install_directory>/etc/collector/server/configs/` are overwritten.
 - Do not enter a directory path because the Load feature only accepts configuration files located in the default path.
 - Do not enter an extension since `.db` is automatically added to the files.

Best practice is to use unique file names to preserve versions, and to use descriptive file names to make the configurations easily identifiable.

Related Topics

- Global collection rules
 - [Node Discovery](#)
 - [Node Access](#)
 - [Node Inclusion](#)
- Per-node rules that override the global rules
 - [Node List](#)
- [What to Collect](#)



Advanced WAE Collector 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 WAE Collector configuration chapters include the following.

- [Snapshot Files](#)—Defines the snapshot process, including how to configure the snapshot files for online network discovery. This chapter also describes differences in the snapshot files used for augmented and manual methods.
- [Augmented Collection](#)—Instructional steps for collecting network data using the augmented method.
- [Manual Collection](#)—Instructional steps for collecting network data using the manual method. This includes several examples, including how to collect BGP LS.
- [Continuous Polling and PCEP LSP Collection](#)—Instructional steps for collecting network data using the manual collection method and the WAE Network Interface (NI) server. This is specific to using continuous polling and/or continuous collection of PCEP LSPs.
- [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 datastore.
- [Flow Collection](#)—Describes the tools used to collect and aggregate exported NetFlow and related flow measurements.
- [Hardware Inventory Collection](#)—Describes how to collect inventory data for use in WAE Live and how to edit default inventory tables. It also describes the process of how inventory is collected.
- [Offline Discovery](#)—Describes the tools used to discover and retrieve information from router configurations and from RRD tools.
- [SAM Integration](#)—Describes integration tools for the SAM application, as well as how to configure the SAM server for use with WAE Collector 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

Advanced configuration chapters use the following terms.

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

Related Topics

- [WAE Collector Overview](#)
- [WAE Collector UI Overview](#)
- *WAE Design Plan Table Schema and CLI Reference*



Snapshot Files


Note

Configuring snapshot files is applicable to both augmented and manual collection methods.

For online discovery, a CLI `snapshot` tool is provided that creates a snapshot of a network at a point in time and saves it to a plan file. A set of files guide this process, including authorization, network access, and snapshot configuration files. This discovery process works well as a scheduled task that captures and stores snapshots of the network.

Both the augmented and manual collection methods use a *snapshot* process to create a plan file that is stored in an archive and available for use by the applications. For the augmented method, the snapshot process adds supplementary data to the plan file created through the WAE Collector UI. For the manual method, the snapshot process is the sole means of network discovery. To determine if you should use either of these methods, or if you should bypass snapshots and use only the WAE Collector UI for collection purposes, refer to the [WAE Collector Overview](#) chapter.

The recommended snapshot workflow is to first obtain the IGP information and set up an organized plan file. Only after that is working satisfactorily is it recommended to incorporate BGP and MPLS information, as both of these add considerable complexity.

Snapshot Configuration Files

The snapshot files consist of a .txt and an optional .inc file, and both are located in `$CARIDEN_HOME/etc` directory. These files are created to meet the basic needs of augmented and manual collections. Refer to [Table 17-1](#) to determine which set of default snapshot files are appropriate for you to use.


Note

To simplify references, this chapter references `snapshot.txt` and `snapshot.inc` files except where a distinction needs to be made.

Table 17-1 **Default Snapshot File Names**

Augmented Snapshot Files	Manual Snapshot Files
<code>snapshot_augment_collector.txt</code>	<code>snapshot.txt</code>
<code>snapshot_augment_collector.inc</code>	<code>snapshot.inc</code>

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. It is these tasks and their `.inc` definitions that 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 `snapshot.inc` file, thus removing the need to manually update these variables more than one time should you reconfigure your network.

- Tasks in the `snapshot.txt` file are defined in the `snapshot.inc` file. These tasks 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.

Recommendation: Make a copy of the default snapshot files, and use the new files for your customizations.

snapshot.txt

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 the [Environment Variables](#) section).
- 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>	Defines whether to collect NetFlow data and related flow measurements.
<ANALYSIS_TASKS>	<ul style="list-style-type: none"> • 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 of these tasks are customized and defined in the `snapshot.inc` file through a series of CLI tools. For information, see the [snapshot.inc](#) section. 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 17-1 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 do 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 17-2). The parameters used for the CLI tools within the tasks are listed in an associated options table (`<options-name>` in Table 17-3).

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.

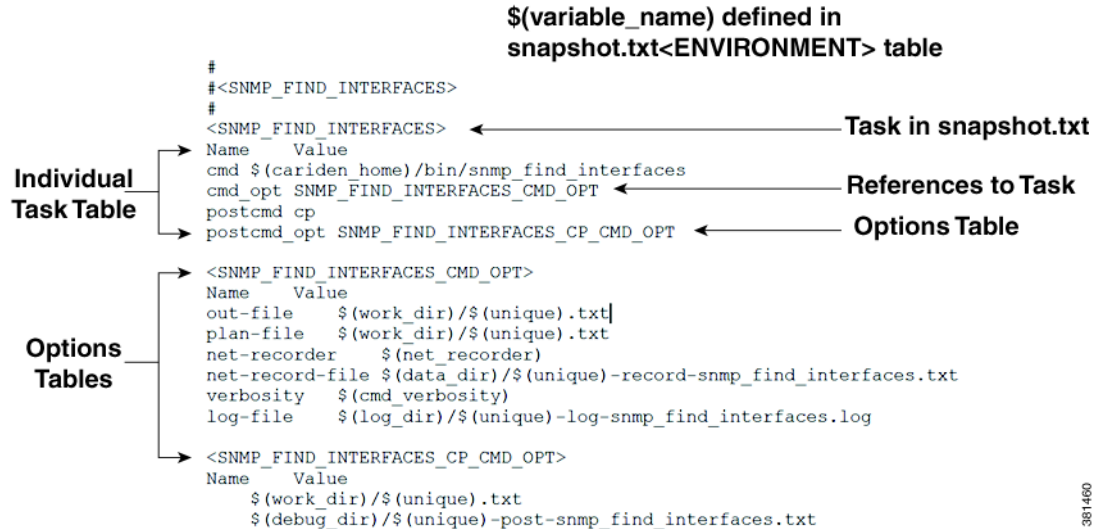
Table 17-2 Individual Task Table

Name	Value
<code>cmd</code>	Fully-qualified name and path of the CLI command to execute. You do not need to change this command during customization.
<code>cmd_opt</code>	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.
<code>cmd_success</code>	Determines what exit codes constitute a success for the command. The snapshot process terminates if the command is unsuccessful. 0 = successful, and 1 = unsuccessful
<code>precmd</code>	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.
<code>precmd_opt</code>	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.
<code>postcmd</code>	Fully-qualified name and path of a command to execute after the CLI command. The default is a Linux <code>cp</code> command that copies intermediate files to the debug directory (<code>debug_dir</code>).
<code>postcmd_opt</code>	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 17-3 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 file names. Example: \$(work_dir)/\$(unique).txt

Figure 17-1 Example Task Defined in snapshot.inc



Launch and Validate snapshot

The `snapshot` tool is located in the `$(CARIDEN_HOME)/bin` directory. You can launch `snapshot` manually or schedule it for periodic operation with a `cron` job. The usual process is to create a `$(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/bugs` 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 time-outs 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 `$(CARIDEN_ROOT)/logs` directory, for example, using `grep`.
- Check the `$(CARIDEN_ROOT)/work` directory to verify the plan file was created.

- Open the plan file in the WAE Design GUI.

Related Topics

- [WAE Collector Overview](#) chapter
- [Augmented Collection](#) chapter
- [Manual Collection](#) chapter
- [Continuous Polling and PCEP LSP Collection](#) chapter
- [Flow Collection](#) chapter
- [Hardware Inventory Collection](#) chapter
- [Multi-Network Collection](#) chapter
- [Network Authentication](#) chapter
- [Network Access File](#) chapter
- [Manage Archives](#) chapter
- *WAE Design Plan Table Schema and CLI Reference*
- *WAE Design Integration and Development Guide*



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, refer to the [WAE Collector Overview](#) chapter.

The process begins by starting and configuring the Collector server to run collections, and this server must continue to run. Optionally, you can start the WAE Network Interface (NI) server and then connect to it from the WAE Collector UI. If you do, then both the Collector server and WAE NI server must continue to run.

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

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, refer to the [Snapshot Examples](#) section in the [Manual Collection](#) chapter.

Parameters

When using the augmentation method, you need to be aware of these parameters.

- 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 authentication.
- 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 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.

This chapter references the following terms.

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

-
- Step 1** Best practice: Back up all configuration files before you begin.
 - Step 2** [Configure the Server](#).
 - Step 3** [Configure Credentials](#).
 - Step 4** 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 5** [Configure Augmented Snapshot Files](#).

Configure the Server

-
- Step 1** Start the web server and access the WAE Collector UI.
 - If it is not running, start the web server.


```
service wae-web-server start
```
 - Access the UI: `https://<Collector_server_IP>:8443/#collector`
 - Log in to the UI.


```
default username: admin
default password: cariden
```
 - Step 2** From the WAE Collector UI, configure the node list and what you want to collect. For a workflow of these steps, see the [WAE Collector UI Overview](#) chapter.
 - Step 3** If continuously polling for traffic statistics or discovering PCEP LSPs, follow these steps.
 - a. If it is not running, start the WAE NI server.


```
service wae-ni start
```
 - b. Configure continuous polling and/or continuous PCEP LSP collection as described in the [WAE Network Interface Server](#) chapter.



Note

One complete collection must occur before continuing.

Configure Credentials

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. Alternatively, you need to authenticate the WAE Network Interface (NI) server if using it for continuous PCEP collection or continuous polling.

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 `$CARIDEN_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 $CARIDEN_ROOT/etc/creds.enc
```



Note

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

Pre-Snapshot Configuration

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 the [Network Authentication](#) chapter.

Step 2 Optional: Customize network access. For information, see the [Network Access File](#) chapter.

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

```
cp $CARIDEN_HOME/etc/snapshot_augment_collector.txt $CARIDEN_ROOT/etc
```

```
cp $CARIDEN_HOME/etc/snapshot_augment_collector.inc $CARIDEN_ROOT/etc
```

If this is not a new installation, you can use existing augmented snapshot files in `$CARIDEN_ROOT/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 the [Snapshot Files](#) chapter.



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, refer to the [Flow Collection](#) chapter.
 - `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** 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 3** 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 the [Snapshot Files](#) chapter.
- 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 the [Configure Credentials](#) section).

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 $CARIDEN_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 $CARIDEN_ROOT/archives/<unique>-archive
-template-dir $CARIDEN_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 $CARIDEN_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 $CARIDEN_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. This instruction assumes you used /opt/cariden as the default installation directory. If you did not, then substitute your installation directory for /opt/cariden.

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
```

Using Collections

- Step 1** If using WAE Live, configure it to collect from the appropriate source and specify the Map archive location. These are set on the WAE Live->Settings->General page. For information, see the *WAE Live Configuration Guide*.
- Step 2** Use the WAE Design GUI to update a template for use by the applications. Use the File->Open From menu to open the template so that you can add visual elements to it. Then use the File-> Save To menu to save it back to the server. For information, refer to the *WAE Network Visualization Guide*.
- Step 3** To verify the plan file collection has properly been set up, open the plan file from the application you are using.

Related Topics

- [WAE Collector UI Overview](#) chapter

- [Snapshot Files](#) chapter
- [Flow Collection](#) chapter
- *WAE Design Plan Table Schema and CLI Reference*
- *WAE Live Configuration Guide*
- *WAE Design Archive User and Administration Guide*



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, refer to the [WAE Collector Overview](#) chapter.

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`.

Workflow

- Step 1** Best practice: Back up all configuration files before you begin.
- Step 2** Execute [Pre-Snapshot Configuration](#) steps, which include creating an authentication file, optionally editing the network access file, and copying the snapshot files for later use.
- Step 3** [Modify Snapshot Files](#).
- Step 4** [Initialize Archive, Create Template, Run Collections](#).
- Step 5** If using the data in applications, execute the [Using Collections](#) steps.

Pre-Snapshot Configuration

- 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 the [Network Authentication](#) chapter.

Step 2 Optional: Customize network access. For information, see the [Network Access File](#) chapter.

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

```
cp $CARIDEN_HOME/etc/snapshot.txt $CARIDEN_ROOT/etc
```

```
cp $CARIDEN_HOME/etc/snapshot.inc $CARIDEN_ROOT/etc
```

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

Modify 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 the [Snapshot Examples](#) section.

For instructions specific to collecting flow data or SAM data, refer to the [Flow Collection](#) and [SAM Integration](#) chapters, respectively.

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.

- ARCHIVE_INSERT—Insert the completed plan file into an external plan file archive that can be accessed by all the applications. For an example, see the [Insert Data into External Archive](#) example.
- ML_INSERT—Manually insert data into the WAE Live datastore. For an example, see the [Insert Data into Datastore](#) example.
- 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. For an example, see the [Insert Data into Map Archive](#) example.

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. For an example, see the [Insert Data into Map Archive](#) example.
- For information on any tool, refer to its `-help` output. For information on how to edit the `snapshot.inc`, see the [Snapshot Files](#) chapter.

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 $CARIDEN_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 $CARIDEN_ROOT/archives/<unique>-archive
-template-dir $CARIDEN_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 $CARIDEN_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 $CARIDEN_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. This instruction assumes you used /opt/cariden as the default installation directory. If you did not, then substitute your installation directory for /opt/cariden.

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
```

Using Collections

- Step 1** If using WAE Live, configure it to collect from the appropriate source and specify the Map archive location. These are set on the WAE Live->Settings->General page. For information, see the *WAE Live Configuration Guide*.
- Step 2** Use the WAE Design GUI to update a template for use by the applications. Use the File->Open From menu to open the template so that you can add visual elements to it. Then use the File-> Save To menu to save it back to the server. For information, refer to the *WAE Network Visualization Guide*.
- Step 3** To verify the plan file collection has properly been set up, open the plan file from the application you are using.

Snapshot Examples

**Note**

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.

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.

<ARCHIVE_INSERT>

Name	Value
<code>cmd</code>	<code>\$(cariden_home)/bin/archive_insert</code>
<code>cmd_opt</code>	<code>ARCHIVE_INSERT_CMD_OPT</code>

<ARCHIVE_INSERT_CMD_OPT>

Name	Value
<code>plan-file</code>	<code>\$(work_dir)/\$(unique).pln</code>
<code>archive</code>	<code>\$(archive_dir)/\$(unique)-archive</code>
<code>time</code>	<code>\$(start_time)</code>

Collect 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.

Step 1 Configure OSC to collect BGP LS. For information, see their *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.

- 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.

<FIND_BGPLS>	
Name	Value
cmd	\$(cariden_home)/bin/find_bgpls
cmd_opt	FIND_BGPLS_CP_CMD_OPT
<SNMP_FIND_BGPLS_CMD_OPT>	
Name	Value
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
```

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 datastore.

- 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`.

<SNMP_FIND_NODES>	
Name	Value
cmd	\$(cariden_home)/bin/snmp_find_nodes
cmd_opt	SNMP_FIND_NODES_CMD_OPT
<SNMP_FIND_NODES_CMD_OPT>	
Name	Value
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>
Name	Value	Value
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>
Name	Value	Value
perf-data	true	true

Step 4 If analyzing LAGs in WAE Live, set the `snmp_find_interfaces -lag` option to true. See the [Collect LAG Membership and Traffic](#) section.

Manually Insert WAE Live Data

This example shows how to insert data directly into the WAE datastore and Map archive, rather than storing on a server or in an external archive.

Insert Data into Datastore

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 datastore during the collection process.

<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)

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.

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

<SNMP_FIND_INTERFACES>

Name	Value
<code>cmd</code>	<code>\$(cariden_home)/bin/snmp_find_interfaces</code>
<code>cmd_opt</code>	<code>SNMP_FIND_INTERFACES_CMD_OPT</code>

<SNMP_FIND_INTERFACES_CMD_OPT>

Name	Value
<code>lag</code>	<code>true</code>
<code>lag-port-match</code>	<code>complete</code>

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.

<SNMP_POLL>

Name	Value
<code>cmd</code>	<code>\$(cariden_home)/bin/snmp_poll</code>
<code>cmd_opt</code>	<code>SNMP_POLL_CMD_OPT</code>

<SNMP_POLL_CMD_OPT>

Name	Value
<code>poll-function</code>	<code>interfaces, ports</code>
<code>polling-interval</code>	<code>interfaces=60</code>
<code>number-of-samples</code>	<code>interfaces=1</code>

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.

<FIND_BGP>	
Name	Value
cmd	\$(cariden_home)/bin/find_bgp
cmd_opt	FIND_BGP_CMD_OPT
<FIND_BGP_CMD_OPT>	
Name	Value
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.

<SNMP_POLL>	
Name	Value
cmd	\$(cariden_home)/bin/snmp_poll
cmd_opt	SNMP_POLL_CMD_OPT
<SNMP_POLL_CMD_OPT>	
Name	Value
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 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.

<SNMP_FIND_NODES>	
Name	Value
cmd	\$(cariden_home)/bin/snmp_find_nodes
cmd_opt	SNMP_FIND_NODES_CMD_OPT

<SNMP_FIND_NODES>	
<SNMP_FIND_NODES_CMD_OPT>	
Name	Value
read-qos-queues	true

Step 3 In `snapshot.inc`, use `snmp_poll` to poll all interface queues with the `-qos-queues '*'` option.

<SNMP_POLL>	
Name	Value
cmd	\$(cariden_home)/bin/snmp_poll
cmd_opt	SNMP_POLL_CMD_OPT
<SNMP_POLL_CMD_OPT>	
Name	Value
poll-function	interfaces
polling-interval	interfaces=60
number-of-samples	interfaces=1
qos-queues	'*'

Related Topics

- [Snapshot Files](#) chapter
- [Continuous Polling and PCEP LSP Collection](#) chapter
- [Flow Collection](#) chapter
- [Hardware Inventory Collection](#) chapter
- [Multi-Network Collection](#) chapter
- [SAM Integration](#) chapter
- [Network Authentication](#) chapter
- [Network Access File](#) chapter
- [Manage Archives](#) chapter
- *WAE Design Plan Table Schema and CLI Reference*
- *WAE Live Configuration Guide*
- *WAE Design Archive User and Administration Guide*



Continuous Polling and PCEP LSP 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 PCEP 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

You cannot use `sam_getplan` when using the WAE NI server.



Note

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, refer to the [Augmented Collection](#) chapter instead of this one.

Workflow

- Step 1** Best practice: Back up all configuration files before you begin.
- Step 2** Set up the WAE NI server.
 - [Configure Continuous Parameters](#)
 - [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](#).
 - [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, then you can 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 [Using Collections](#) steps.

Set Up WAE NI Server

Configure Continuous Parameters

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

**Note**

If you are using the Collector server, do not edit this file.

Continuously Collect PCEP LSPs

**Note**

PCEP LSPs are managed by WAE. LSPs that are managed by WAE (PCE) 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.)

Parameter	Description
<code>enablePcepLspCollection</code>	True = Continuously collect PCEP LSPs. False = Do not continuously collect PCEP LSPs

Continuously Poll Traffic

**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.
delegateStatsToContinuousPoller	True = Run continuously polling. False = Do not run continuously polling.
rawCounterTtlInMins	Defines the amount of time raw counters are kept in minutes. The minimum value is 5 minutes.

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 <code>collector_getplan</code> 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. For information on changing the password, contact your support representative.
- 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.

-
- 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 the [Network Authentication](#) chapter.
- Step 2** Optional: Customize network access. For information, see the [Network Access File](#) chapter.

Step 3 Since you need to run two snapshots, create both sets of files now. You will later be editing 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 `$CARIDEN_ROOT/etc` and **give them different names**.

Note that if you have existing snapshot files in `$CARIDEN_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 $CARIDEN_HOME/etc/snapshot.txt $CARIDEN_ROOT/etc/snapshot-pushplan.txt
cp $CARIDEN_HOME/etc/snapshot.inc $CARIDEN_ROOT/etc/snapshot-pushplan.inc
cp $CARIDEN_HOME/etc/snapshot_augment_collector.txt
  $CARIDEN_ROOT/etc/snapshot-getplan.txt
cp $CARIDEN_HOME/etc/snapshot_augment_collector.inc
  $CARIDEN_ROOT/etc/snapshot-getplan.inc
```

Create Snapshot to Push Plan Files



Note

For information on configuring `snapshot.txt` and `snapshot.inc` files, see the [Snapshot Files](#) chapter. 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 $CARIDEN_ROOT/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 the [Configure Push Credentials](#) section).

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

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

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

<COLLECTOR_PUSHPLAN>	
jms-server-port	61617
<COLLECTOR_PUSHPLAN_CP_CMD_OPT>	
Name	Value
	\$(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 the [Run Collections](#) section.

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 $(CARIDEN_ROOT)/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



Note

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, refer to the [Snapshot Examples](#) section in the [Manual Collection](#) chapter.

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 all the 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 the [Configure Get Credentials](#) section).

The `-credentials-file` must match the name that you specified when you first set the credentials (as per the [Configure Get Credentials](#) section), 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

<COLLECTOR_GETPLAN>	
Name	Value
<code>cmd</code>	<code>\$(cariden_home)/bin/collector_getplan</code>
<code>cmd_opt</code>	<code>COLLECTOR_GETPLAN_CMD_OPT</code>
<code>postcmd</code>	<code>cp</code>
<code>postcmd_opt</code>	<code>COLLECTOR_GETPLAN_CP_CMD_OPT</code>
<code>cmd_success</code>	<code>0</code>
<COLLECTOR_GETPLAN_CMD_OPT>	
Name	Value
<code>set-credentials</code>	<code>false</code>
<code>credentials-file</code>	<code>\$(home_dir)/etc/collector/credentials-CP.enc</code>
<code>get</code>	<code>files</code>

<COLLECTOR_GETPLAN>	
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>	
Name	Value
	\$(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 $CARIDEN_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 $CARIDEN_ROOT/archives/<unique>-archive
-template-dir $CARIDEN_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 $CARIDEN_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.

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 $CARIDEN_ROOT/etc/snapshot-pushplan.txt
```

```
snapshot -config-file $CARIDEN_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`. This instruction assumes you used `/opt/cariden` as the default installation directory. If you did not, then substitute your installation directory for `/opt/cariden`.

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
```

Using Collections

- If using WAE Live, configure it to collect from the appropriate source and specify the Map archive location. These are set on the WAE Live->Settings->General page. For information, see the *WAE Live Configuration Guide*.
- Use the WAE Design GUI to update a template for use by the applications. Use the File->Open From menu to open the template so that you can add visual elements to it. Then use the File-> Save To menu to save it back to the server. For information, refer to the *WAE Network Visualization Guide*.
- To verify the plan file collection has properly been set up, open the plan file from the application you are using.

Related Topics

- [Snapshot Files](#) chapter
- [Network Authentication](#) chapter
- [Network Access File](#) chapter
- *WAE Design Plan Table Schema and CLI Reference*
- *WAE Live Configuration Guide*



Multi-Network Collection



Note

Running multi-network collections is supported only for the manual collection method, and is only supported in the WAE Live application.

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 the [Snapshot Files](#) chapter and [Manual Collection](#) chapter, respectively.

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 the [Network Authentication](#) and [Network Access File](#) chapters.

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 $CARIDEN_HOME/etc/snapshot.txt $CARIDEN_ROOT/etc/ss-east.txt
cp $CARIDEN_HOME/etc/snapshot.inc $CARIDEN_ROOT/etc/ss-east.inc
cp $CARIDEN_HOME/etc/snapshot.txt $CARIDEN_ROOT/etc/ss-north.txt
cp $CARIDEN_HOME/etc/snapshot.inc $CARIDEN_ROOT/etc/ss-north.inc
```

If this is not a new installation, you can use existing snapshot files in `$CARIDEN_ROOT/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 $CARIDEN_ROOT/archives/<unique>-archive
```

Example: `archive_init -archive $CARIDEN_ROOT/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`. This instruction assumes you used `/opt/cariden` as the default installation directory. If you did not, then substitute your installation directory for `/opt/cariden`.

- 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.

- Step 5** On the WAE Live Settings->General Settings page **follow these steps for each network**. These steps describe only the nuances of configuring collection for a specific network using external archives. For more information, see the *WAE Live Configuration Guide*.
- a. Create a unique network to identify the network from which you are collecting. This does not have to be identical to the archive, but best practice is to use a name that identifies the network. To enable collection, this must be set to Active.
 - b. Select the newly added network.
 - c. Set the Collection Source to External Archive. Text in <angle brackets> refers to environment variables that you set in the `ss-east.txt` file. This must match the path entered for `archive_init`.
Example: `/opt/cariden/archives/east-archive`
 - d. If using the Map component, set a unique Map Archive directory. This is the local directory into which collected plan files and the template file are stored for display in the Map. This directory must be a different location than that of the external archive. Best practice is to use a name that identifies the network.
Example: `/opt/cariden/data/mldata/archive/east`

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 datastore. 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 datastore 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 `$(CARIDEN_ROOT)/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_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 datastore.

**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. This instruction assumes you used /opt/cariden as the default installation directory. If you did not, then substitute your installation directory for /opt/cariden.

- 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
```

Step 5

On the WAE Live Settings->General Settings page **follow these steps for each network**. These steps describe only the nuances of configuring collection for a specific network using manually inserted plan files. For more information, see the General Settings chapter in the *WAE Live Configuration Guide*.

- a. Create a network by that same, case-sensitive name as used in the `ml_insert_plan -network` option. To enable collection, this network must be set to Active.

Example: north

- b. Select the newly added network.

- c. Set the Collection Source to None.

- d. If using the Map component, set a unique Map archive directory. This is the local directory into which collected plan files and the template file are stored for display in the Map. This must match the case-sensitive path used in the MAP_ARCHIVE_INSERT task (that calls `archive_insert`).

Example: /opt/cariden/data/mldata/archive/north

Related Topics

- [Snapshot Files](#) chapter
- [Manual Collection](#) chapter
- *WAE Live Configuration Guide*
- *WAE Design Plan Table Schema and CLI Reference*



Flow Collection



Note

Collecting NetFlow and related flow measurements is supported in both the augmented and manual configuration methods. The recommendation is to use the augmented method if it supports the network configuration.

WAE Collector can collect and aggregate exported NetFlow and related flow measurements. These may be used to construct accurate demand traffic data for both WAE Design and WAE Live. Flow collection provides an alternative to the estimation of demand traffic from interface, LSP, and other statistics using Demand Deduction. 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 AS's is of interest, for example when tracking demands over time in WAE Live.

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

Flows imported into WAE Collector are also used in the Business Intelligence (BI) solutions.

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

- NetFlow, JFlow, CFlowd, and IPFIX flows
- SNMP
- BGP path attributes

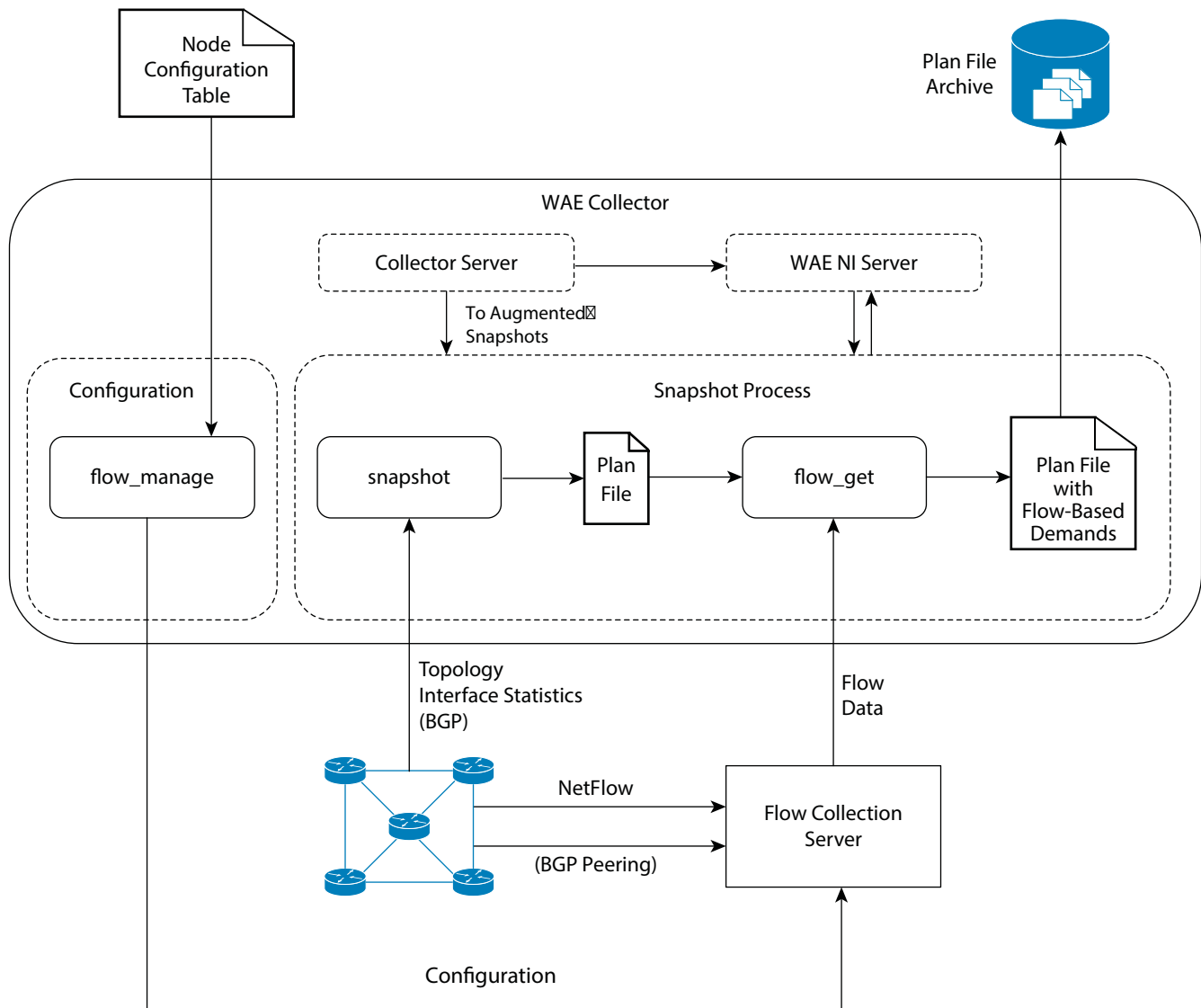
This chapter references the following terms.

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

Figure 22-1 shows the workflow for collecting flow data from the external network and flow collection server. The WAE Collector tools, `flow_manage` and `flow_get`, integrate with an external configuration table and the snapshot process, respectively. The end result is a plan file containing flow-based demands.

Figure 22-1 Flow Collection Workflow



- `<NodeFlowConfigs>` table—This user-created table contains router information that is used as input to `flow_manage`. At minimum, it identifies node names, sampling rates, and the source IP of the flows. See the [<NodeFlowConfigs> Table](#) section.
- `flow_manage`—This CLI tool enables network connectivity and manages the data collection, including starting and stopping the flow collection process. It uses input from the `<NodeFlowConfigs>` table to generate configuration information, which it then sends to the flow

collection server. This CLI tool also starts and stops the flow collection server. The `flow_manage` tool must be invoked outside the snapshot files. See the [flow_manage](#) section and `flow_manage -help` output.

- Flow collection server—The background process receiving configuration information from `flow_manage`, which it uses to establish network connectivity and receive flow data and BGP attributes. The server then correlates this data and forwards it to the `flow_get` tool.
- `flow_get`—This CLI tool, which is enabled inside the snapshot files, gets the flow data from the flow collection server and puts it into the plan file that was generated as part of the snapshot process. See the [flow_get](#) section and `flow_get -help` output.

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. Following are a few 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 and/or border routers. If this is not feasible, then 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.
- 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).

For additional information, refer to configuration examples on <http://wiki.pmacct.net>.

Manage Flow Collection

<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-delimited format.
- Include one row per node (router) from which you are collecting flow data.
- Enter contents described in [Table 22-1](#) for each of these nodes. The BGP columns are required only if collecting BGP information. [Table 22-2](#) provides an example.

By default, the flow collection server looks for this file in the `$CARIDEN_HOME/lib/ext/pmacct` directory. You can, however, specify another location and identify the full path in the `flow_manage` tool.

Table 22-1 <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 <code>flow_manage -bgp</code> option is true.
BGPPassword	BGP peering password for MD5 authentication. Use this column if the <code>flow_manage -bgp</code> option is true and if BGPSourceIP has a value.

Table 22-2 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::8 a4e: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

flow_manage

The `flow_manage` tool starts and stops 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. The following table lists the options available for `flow_manage`.

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.

```
flow_manage -server-ip 192.168.1.3 -action reload -node-flow-configs-table
flowconfigs.txt
```

Get Traffic Matrices into a Plan File

Unlike other traffic, this flow collection process creates and puts demands and its traffic directly into the plan file using `flow_get`, which is executed within the snapshot process. The resulting plan file includes a traffic matrix, or `<DemandTraffic>` table, with a set of demands between all nodes, between nodes in an internal AS and its peering AS's, or between external AS's, depending on the `flow_get` option selected.

Demands

When the `flow_get -demands` option is true, demands are appended to the plan and named with the prefix `flow`. If a demand with the same name and key properties already exists, it is given a different, sequentially-numbered name starting with 2 (`flow[2]`).

- If the plan file contains no external AS's, then demands are created between nodes.
- If the plan file contains external AS's, then the demand sources and destinations are assigned to external AS nodes in the plan file.
- If the external AS's cannot be identified, the sources and destinations are set to the node containing the external interface that is the ingress or egress interface for the flow.

Demands are sourced from and destined for either nodes or AS's via their node. For instance, if a demand originates from a node that is internal to the plan file or is in a neighboring AS that is not in the plan file, then the source is the node at which the traffic enters the plan file.

Whether or not the resulting source or destination is a node or AS depends on whether the neighboring source AS or destination AS is in the plan file. If the neighboring source AS is an external AS, then the source of demands is the node in the AS through which traffic flows.

Demand Traffic

The demand traffic is constructed from the collected flow information input into the `<DemandTraffic>` table. Additionally, `flow_get` populates three related columns in the `<NetIntInterfaces>` table, as follows.

- `FlowTrafficEstimate`—A T (true) or F (false) identifies whether the interface traffic is estimated because there are multiple AS's connected by this interface or because the neighboring AS cannot be resolved.
- `FlowTraffic`—Sum of the flows on the interface.

- **FlowTrafficRatio**—Multiple flows and their traffic entering the network on each interface are collected. The flow traffic is scaled upwards to match the SNMP traffic measurements of each ingress interface. The calculation is as follows.

(SNMP Measurement of the Interface / Sum of Measured Flow Traffic over Interfaces)

Example: If the SNMP measurement for an interface is 500 Mbps, and if two flows are on that interface, each measuring 200 Mbps, the FlowTrafficRatio for that interface would be as follows.

$(500 / (200 + 200) = 1.25$

If a node in an internal AS has multiple connections (ports) connecting to a peer AS and to multiple destinations within the peer AS, WAE Collector aggregates the flow traffic, as well as determines port-to-port traffic.

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.

To execute the `flow_get` CLI tool, you must define the input and output plan file name. All other parameters are optional.

Example: The following command gets the data from the `/acme/infile.txt` file, adds a demand traffic matrix to it, and outputs it to the `/acme/outfile.txt` file. If any external BGP interfaces are detected as missing flow data, this information is written to the `/acme/ext_no.txt` file. Interfaces from which flow data was received are also marked in the `<Interfaces>` table.

```
flow_get -plan-file /acme/infile.txt -out-file /acme/outfile.txt -demands true
-missing-flows /acme/ext_no.txt
```

Examples: Create a list of demands for IPv4 and IPv6.

Concurrent: `flow_get -plan-file /acme/infile.txt -out-file /acme/outfile.txt -demands true -address-family ipv4,ipv6`

Aggregated: `flow_get -plan-file /acme/infile.txt -out-file /acme/outfile.txt -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.txt -out-file /acme/outfile.txt
-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 discovered. 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 blacklists or whitelists of nodes and then passing these tags to `flow_get -ext-node-tags`. Interfaces connected to a node matching one or more of the tagged nodes are marked as excluded, and measurements received by these nodes are discarded.

Step 1 Tag the nodes that 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.txt -out-file /acme/outfile.txt -ext-node-tags non_core
```

Collect Flows

Step 1 If using the augmented method of collection, you must be running the Collector server and/or the WAE Network Interface (NI) server.

Optional: Configure the server to collect BGP peering information if using augmented snapshots. For information, see the [WAE Collector UI Overview](#) chapter.

Step 2 Create the <NodeFlowConfigs> table. See the <NodeFlowConfigs> [Table](#) section.

Step 3 Execute `flow_manage` to start the flow collection server. See the [flow_manage](#) section and `flow_manage -help` output.

Step 4 Configure the snapshot files to execute the appropriate snapshot tasks, including `flow_get`. See the [flow_get](#) section, [Snapshot Integration](#) section, and `flow_get -help` output

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 ([Figure 22-1](#)). 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).

Note that while many tasks are optional, the following sequence includes FIND_BGP and TRIM_NODES since they are commonly used.

**Note**

If combining with `collector_getplan` with `flow_get`, you must execute `flow_get` after `collector_getplan`.

Task	snapshot_augment_collector.txt (Augmented Collection)	snapshot.txt (Manual Collection)
SNMP_FIND_INTERFACES		x
FIND_BGP (optional)		x
SNMP_POLL		x
DMD_MESH_CREATOR (when there is incomplete NetFlow coverage and there is a need to run Demand Deduction)	x	x
FLOW_GET	x	x
TRIM_NODES (optional)	x	x
DMD_DEDUCT (when there is incomplete NetFlow coverage and there is a need to run Demand Deduction)	x	x

When you call FIND_BGP, the following information is collected.

- Ingress peer AS and associated interface information for IPv4 and IPv6
- eBGP multihop peer interface information

Collection of both ingress and egress peer AS information makes it possible for WAE Collector to accurately characterize BGP routing asymmetry.

The `snapshot_augment_collector.inc` and `snapshot.inc` files includes all the necessary CLI tools for collecting flow data. If needed, change the tools' definitions to meet your flow collection needs.

Related Topics

- [Snapshot Files](#) chapter
- [Augmented Collection](#) chapter
- [Manual Collection](#) chapter
- *WAE Design Plan Table Schema and CLI Reference*



Hardware Inventory Collection

The inventory collection process is based on two tools that collect and process the network hardware to create the NetIntNodeInventory table used by WAE Live to produce inventory reports.

- `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.
- `build_inventory`—Processes the raw hardware data information in the NetIntHardware* tables) to categorize and remove unwanted objects in the final NetIntNodeInventory table.

This chapter is divided into two areas.

- [Configure Inventory Collection](#)—Configure inventory collection using snapshot files. You can use the default inventory configuration files or edit them prior to following these steps.
- [Edit Default Inventory Files](#)—Make changes to default inventory files for use in inventory collection. Note that this section provides background on how inventory collection works.

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`.

Configure Inventory Collection

- Step 1** Best practice: Back up all configuration files before you begin.
- Step 2** Execute [Pre-Snapshot Configuration](#) steps, which include copying the inventory configuration files for later use and setting credentials.
- Step 3** [Configure Snapshot Files](#).
- Step 4** [Test and Run Collections](#).



Note For ease of understanding, these instructions reference `snapshot_inventory.txt` and `snapshot_inventory.inc` files.

Pre-Snapshot Configuration

Step 1 For new installations, copy the default `snapshot.txt` and `snapshot.inc` files to working inventory configuration files.

```
cp $CARIDEN_HOME/etc/snapshot.txt $CARIDEN_ROOT/etc/snapshot_inventory.txt
cp $CARIDEN_HOME/etc/snapshot.inc $CARIDEN_ROOT/etc/snapshot_inventory.inc
```

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

Step 2 For new installations, create an `inventory` directory in `$CARIDEN_ROOT/etc`, and then copy the default inventory files to the working directory.

```
cp $CARIDEN_HOME/etc/inventory/master_inventory_templates.txt
$CARIDEN_ROOT/etc/inventory

cp $CARIDEN_HOME/etc/inventory/master_exclude_list.txt $CARIDEN_ROOT/etc/inventory

cp $CARIDEN_HOME/etc/inventory/master_hw_spec.txt $CARIDEN_ROOT/etc/inventory
```

If this is not a new installation, you can use existing snapshot files in `$CARIDEN_ROOT/etc/inventory`, and make modifications to them if needed.

Step 3 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. Alternatively, you need to authenticate the WAE Network Interface (NI) server if using it for continuous PCEP collection or continuous polling.

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

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

The default credential file path, which is configurable, is `$CARIDEN_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
$CARIDEN_ROOT/etc/inv_creds.enc
```



Note

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

Configure Snapshot Files

Step 1 Configure a collection for the same network for which you are gathering inventory data such that the resulting plan file resides on either the Collector server or the WAE Network Interface (NI) server.

Create a network in WAE Live, and configure it to collect plan files for this same network from this server. For information on configuring WAE Live, see the *WAE Live Configuration Guide*.

Step 2 Edit the `snapshot_inventory.txt` file to define environment variables of common values used in the `snapshot_inventory.inc` file.

- At minimum, you must define `unique`, `home_dir`, and `collector_url`,
- 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 http://localhost:8443`

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

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

Step 3 Edit the `snapshot_inventory.txt` file to contain **only** the necessary tasks.

<ul style="list-style-type: none"> • Comment out or remove all tasks except for <code>COLLECTOR_GETPLAN</code> and <code>MATE_CONVERT</code>. • Add the <code>GET_INVENTORY</code> task to create the <code>NetIntHardware*</code> tables. • Add the <code>BUILD_INVENTORY</code> task to process these hardware tables for the creation of the <code>NetIntNodeInventory</code> used by WAE Live. • Add the <code>ML_INSERT_CTL</code> task, which will insert and schedule the insertion of inventory data into the WAE Live datastore. 	<pre><DISCOVERY_TASKS> COLLECTOR_GETPLAN <ANALYSIS_TASKS> MATE_CONVERT GET_INVENTORY BUILD_INVENTORY <ARCHIVE_INSERT_TASKS> ML_INSERT_CTL</pre>
---	---

Step 4 Edit the `snapshot_inventory.inc` file to call all four of the above tasks. For information on any tool, refer to its `-help` output.

- Configure `collector_getplan` tool to call the plan file from the same server for which you collected in step 1.
- Configure `get_inventory` to collect the hardware inventory.

Optional: To allow logging in to the router to collect inventory data, set the `get_inventory -login-allowed` option to `false`. (By default, it is `true`.)

- Configure `build_inventory` to process the hardware inventory to create information that WAE Live can use.

Optional: To broaden the search when processing raw inventory data, set the `build_inventory -guess-template-if-nomatch` option to `true`. (By default, it is `false`.)

If using any inventory configuration files named other than the default names, you must call these options from the `build_inventory` tool.

For information on the `build_inventory` options and the inventory configuration files, see the [Edit Default Inventory Files](#) section.

- Add the `mate_convert` tool to convert the resulting plan from `.txt` to `.pln`.
- Configure `ml_insert_ctl` to schedule the hardware inventory insertion into the Live datastore.

You must specify the `-network` option for the `ml_insert_ctl` tool. This network must match the one configured in WAE Live to use for the topology and statistics collection into which this inventory data is added.

For information on any tool, refer to its `-help` output. For information on how to edit the `snapshot_inventory.inc` file, see the [Snapshot Files](#) chapter.

Example `snapshot_inventory.inc`

<COLLECTOR_GETPLAN>	
Name	Value
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>	
Name	Value
set-credentials	false
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>	
Name	Value
	\$(work_dir)/\$(unique).txt
	\$(debug_dir)/\$(unique)-post-collector_getplan.txt
<GET_INVENTORY>	
Name	Value
cmd	\$(cariden_home)/bin/get_inventory
cmd_opt	GET_INVENTORY_CMD_OPT
postcmd	cp
postcmd_opt	GET_INVENTORY_POST_CMD_OPT
<GET_INVENTORY_CMD_OPT>	
Name	Value
login-allowed	true

plan-file	\$(work_dir)/\$(unique).txt
out-file	\$(work_dir)/\$(unique).txt
<GET_INVENTORY_POST_CMD_OPT>	
Name	Value
	\$(work_dir)/\$(unique).txt
	\$(debug_dir)/\$(unique)-post-get_inventory.txt
<BUILD_INVENTORY>	
Name	Value
cmd	\$(cariden_home)/bin/build_inventory
cmd_opt	BUILD_INVENTORY_CMD_OPT
postcmd	cp
postcmd_opt	BUILD_INVENTORY_POST_CMD_OPT
<BUILD_INVENTORY_CMD_OPT>	
Name	Value
guess-template-if-nomatch	true
plan-file	\$(work_dir)/\$(unique).txt
out-file	\$(work_dir)/\$(unique).txt
<BUILD_INVENTORY_POST_CMD_OPT>	
Name	Value
	\$(work_dir)/\$(unique).txt
	\$(debug_dir)/\$(unique)-post-build_inventory.txt
<MATE_CONVERT>	
Name	Value
cmd	\$(cariden_home)/bin/mate_convert
cmd_opt	MATE_CONVERT_CMD_OPT
postcmd	cp
postcmd_opt	MATE_CONVERT_POST_CMD_OPT
<MATE_CONVERT_CMD_OPT>	
Name	Value
plan-file	\$(work_dir)/\$(unique).txt
out-file	\$(work_dir)/\$(unique).pln
<MATE_CONVERT_POST_CMD_OPT>	
Name	Value
	\$(work_dir)/\$(unique).pln
	\$(debug_dir)/\$(unique)-post-mate_convert.pln

<ML_INSERT_CTL>	
Name	Value
cmd	\$(cariden_home)/bin/ml_insert_ctl
cmd_opt	ML_INSERT_CTL_CMD_OPT
<ML_INSERT_CTL>	
Name	Value
	-insert
time	\$(start_time_direct)
handler	ml_inventory.json
network	default
plan-file	\$(work_dir)/\$(unique).pln

Test and Run Collections

- Step 1** 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 $CARIDEN_ROOT/etc/snapshot_inventory.txt
```

- Step 2** Create a cron job that repeats the process of creating snapshots to collect hardware inventory and inserting them into the WAE Live datastore once a day.



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`. This instruction assumes you used `/opt/cariden` as the default installation directory. If you did not, then substitute your installation directory for `/opt/cariden`.

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
```


Use Inventory in WAE Live

The WAE Live Settings->General Settings page must be configured to collect the plan files containing topology and statistics for the network used in the inventory collection process. WAE Live must get these plan files from the same server as configured in step 1 of the [Configure Snapshot Files](#) section.

Given the inventory collection process gets this same plan file (`collector_getplan`) and inserts the inventory into the same network (`ml_insert_ctl -network`), there are no required inventory configurations on the WAE Live Settings->General Settings page.

**Note**

These networks must be the same: the network for which WAE Live is configured to get plan files and the network into which the inventory is inserted.

For information on multi-network collections, see the [Multi-Network Collection](#) chapter. For information on configuring WAE Live to get collections, see the *WAE Live Configuration Guide*.

Edit Default Inventory Files

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.

- `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.

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 23-1](#).

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.

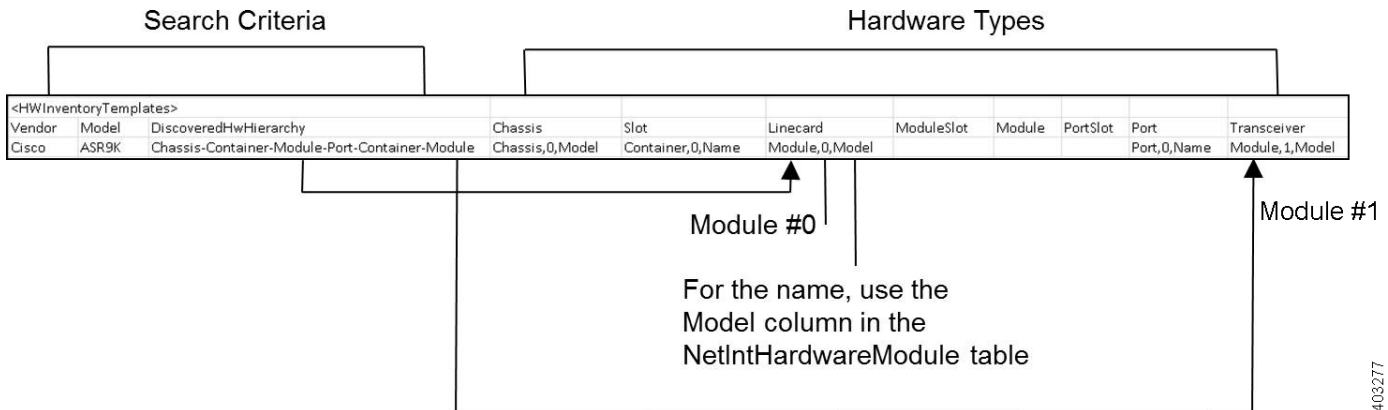
Cisco ASR9K Chassis-Container-Module-Port-Container-Module

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 23-1 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 the [Hardware Hierarchy](#) section.

- 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	\A4Z	'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		\VCPU0/129\$
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

Related Topics

- [Snapshot Files](#) chapter
- [Augmented Collection](#) chapter
- [Manual Collection](#) chapter
- [Continuous Polling and PCEP LSP Collection](#) chapter
- *WAE Design Plan Table Schema and CLI Reference*
- *WAE Live Configuration Guide*



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 the [Import Router Configuration Files](#) section. For information on using this tool from the WAE Design GUI, see the *WAE Design Integration and Development Guide*.
- `parse_igp`—Converts IGP information from router `show` commands to a plan file. See the [Import IGP Database](#) section. For information on using this tool from the WAE Design GUI, see the *WAE Design Integration and Development Guide*.
- `get_show`—A tool for entering router `show` commands.



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.

<ul style="list-style-type: none"> • 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) 	<ul style="list-style-type: none"> • IGP type and metrics (IS-IS or OSPF) <ul style="list-style-type: none"> – Process ID (OSPF) – Instance ID (IS-IS) • RSVP reservable bandwidth (MPLS) • MPLS LSPs (including bandwidth and FRR LSPs) • LAG¹ 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:	<code>terminal length 0</code>
Juniper:	<code>set cli screen-length 0</code>

- 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 run this command.

```
no hostname dynamic
```

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

Cisco:	<code>show isis database verbose</code>
Juniper:	<code>show isis database extensive</code>

- 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
<p>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.</p> <pre data-bbox="381 436 706 514">IOS: ip ospf name-lookup IOS XR: ospf name-lookup</pre> <p>Display the OSPF database.</p> <pre data-bbox="381 619 820 697">IOS: show ip ospf database router IOS XR: show ospf database router</pre> <p>Display the OSPF TE database.</p> <pre data-bbox="381 802 885 879">IOS: show ip ospf database opaque-area IOS XR: show ospf database opaque-area</pre> <p>Display OSPFv3 database (IOS only)</p> <pre data-bbox="381 984 787 1102">show ipv6 ospf database router show ipv6 ospf database link show ipv6 ospf database prefix</pre>	<p>Display the OSPF database.</p> <p>If the system supports DNS, pipe the output of the <code>show</code> command to the resolver so the database has host names, rather than just IP addresses.</p> <pre data-bbox="992 535 1502 562">show ospf database extensive resolve</pre> <p>Otherwise, just show the database, which identifies routers by IP address only.</p> <pre data-bbox="992 703 1372 730">show ospf database extensive</pre> <p>Display OSPFv3 database.</p> <pre data-bbox="992 840 1388 867">show ospf3 database extensive</pre>

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

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.

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 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.

Related Topics

- [Snapshot Files](#) chapter
- *WAE Design Integration and Development Guide*
- *WAE Design Table Plan Schema and CLI Reference*



SAM Integration



Note

Integrating SAM collection into the snapshot process is supported only for the manual collection method. Additionally, you cannot run `sam_getplan` with the WAE Network Interface (NI) server.

WAE Collector includes integration tools for Alcatel-Lucent's 5620 Service Aware Manager (SAM, versions 10.0 to 12.0). The information collected from a SAM server can be integrated into the WAE Collector snapshot process.

This chapter is divided into the following main sections.

- [SAM Discovery](#)—Lists the information that WAE Collector gathers from SAM.
- [Configure SAM for Use with WAE Collector](#)—Describes how to configure the SAM server to collect statistics for use with WAE Collector.
- [SAM Integration](#)—Describes options for retrieving topology and traffic measurements from a network managed by a SAM server.

SAM Discovery

WAE Collector collects the objects, routing and peering information and associated traffic from SAM tools as defined in [Table 25-1](#). WAE Collector does not collect this information for point-to-multipoint (P2MP) LSPs, service classes, Layer 1 topology, or the mapping of interface queues to service classes.

Traffic data collection is possible only through SAM.

Table 25-1 WAE Collector Discovery with SAM Integration

Discovered	Description
Objects	<ul style="list-style-type: none"> • Nodes • Interfaces • Interface Queues (egress only or combined egress and ingress) • RSVP-TE LSPs • FRR LSPs • Shared-risk link groups (SRLGs) • LAG ports • Layer 2 (L2) VPNs (VLL and VPLS) • Layer 3 (L3) VPRNs • T1/E1 bundle ports
Routing and Peering Information	<ul style="list-style-type: none"> • OSPF • IS-IS • RSVP-TE
Traffic	<ul style="list-style-type: none"> • Interfaces • Interface Queues • RSVP-TE LSPs¹ • Service distribution points (SDPs) • L2 and L3 VPN access interfaces • LAGs • T1/E1 bundles
Performance	<ul style="list-style-type: none"> • Interfaces <ul style="list-style-type: none"> – Dropped packets out – Error packets in • Nodes <ul style="list-style-type: none"> – Memory utilization – Route Processor (CPU) utilization

1. You can collect either LSP or SDP traffic statistics, but not both

SDPs

Alcatel-Lucent's VPN service traffic is transported between provider edge (PE) routers by circuits aggregated in unidirectional service tunnels called *SDP bindings*. These SDPs terminate at a destination router, which directs packets to the correct service egress interface on that device.

- SDPs are not used for local services because the same PE router is the source and the destination.
- Each SDP encapsulates the data between the two PE routers and appears as an L2 path to the service traffic, although it is actually traversing an IP or IP/MPLS core.

- WAE Collector does not model SDPs.
- WAE Collector retrieves and stores SDP statistics in place of LSP statistics only when LSP traffic statistics are not available in SAM due to Alcatel-Lucent hardware limitations.

Configure SAM for Use with WAE Collector

Before using WAE Collector discovery tools with SAM, you must configure SAM so that WAE Collector can import the statistics collected. There are four required steps.

-
- Step 1** [Configure SAM Group and User Account with OSS Permission](#) (set up the user account on the SAM server).
 - Step 2** [Configure SAM to Collect and Store Performance Statistics](#).
 - Step 3** [Configure SAM to Collect and Store Accounting Statistics](#).
 - Step 4** [Verify Accounting Statistics Collection](#) for specific policies.

Once you have configured SAM, wait for polling to occur twice. By default the polling time is 15 min, in which case you need to wait at least 30 minutes before loading the plan file.



Note

These instructions are for SAM version 9.0. Although the steps are applicable to all SAM versions, the actual interface options (for example, menus) might be different for other versions.

Configure SAM Group and User Account with OSS Permission

To constrain access to those using WAE and give that group permission to the API, you must create the group and its users, and assign the group an oss permission.

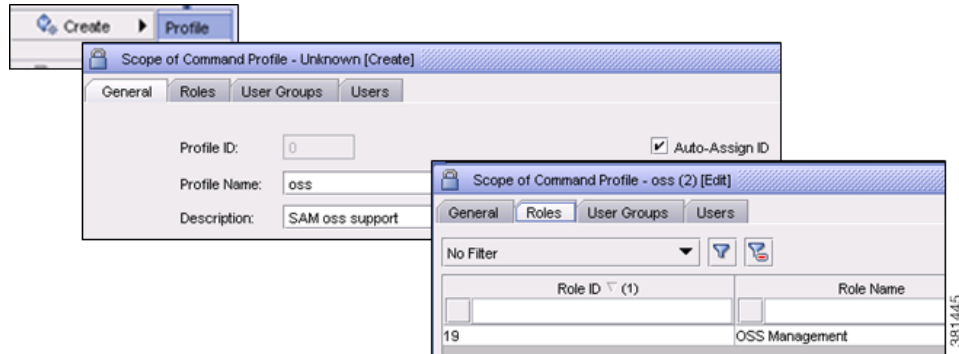
- Log into the SAM server.
- [Create an OSS Permission](#) for a new group that will be using WAE.
- [Create a Group for WAE Users](#) and assign this group the oss permission.
- [Add Users to the WAE Group](#).

Create an OSS Permission

-
- Step 1** Create an oss permission for a new group of users who will be using WAE.
 - Step 2** Select the Administration->Security->5620_SAM User Security menu.
 - Step 3** Select the Scope of Command tab.
 - Step 4** Click Create, and select Profile. The Scope of Command Profile dialog box appears ([Figure 25-1](#)).
 - a. In the Profile Name field, enter oss.
 - b. Optional: Enter a description.
 - c. Select the Roles tab and click Add.
 1. Select OSS Management from the Role Name list.
 2. Click OK to add, and then again to confirm.

3. Click OK to accept the changes and close the Scope of Command Profile dialog box.
- Step 5** If continuing, leave the 5620_SAM User Security dialog box open. Otherwise, click Cancel to close it.

Figure 25-1 Example of Creating an OSS Permission

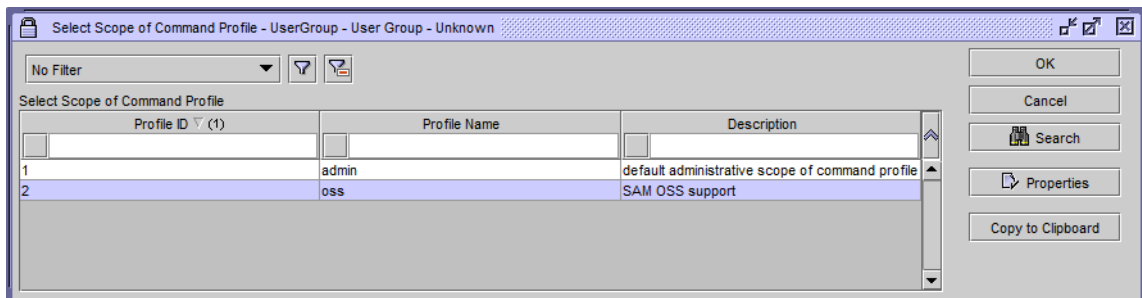


Create a Group for WAE Users

Create a group for using WAE and assign this group the oss role.

- Step 1** Select the Administration->Security->5620_SAM User Security menu.
- Step 2** Select the User Groups tab.
- Step 3** Click Create. The User Group dialog box appears.
- In the User Group name, enter 5 or more characters to identify the group that will be using WAE. For example, wae-oss.
 - Optional: Enter a description.
 - In the Scope of Command area, click Select.
 - Select the oss profile name that you just created (Figure 25-2).
 - Click OK.
 - In the Span of Control area, click Select.
 - Select default as the profile name.
 - Click OK.
 - Click OK to accept the changes and close the User Group dialog box.
- Step 4** If continuing, leave the 5620_SAM User Security dialog box open. Otherwise, click Cancel to close it.

Figure 25-2 Example Scope of Command Entry



Add Users to the WAE Group

Add one or more users to the WAE group.

-
- Step 1** Select the Administration->Security->5620_SAM User Security menu.
- Step 2** Select the Users tab.
- Step 3** Click Create. The User dialog box appears.
- a. In the User Name field, enter the user's name.
 - b. Optional: Enter a description.
 - c. Next to the User Group field, click Select.
 1. Select the group you just created.
 2. Click OK.
 - d. In the Password area, enter your user password in the Password field. Re-enter the password in the Confirm Password field.
 - e. Click OK to accept the changes and close the User dialog box.
- Step 4** Click Cancel to close the 5620_SAM User Security dialog box.

Configure SAM to Collect and Store Performance Statistics

-
- Step 1** Select the Tools->Statistics->MIB Policies menu.
- Step 2** Click Search to view a list of all MIB statistics policies.
- Step 3** Select Default Stats Policy from this list.
- Step 4** Click Properties. The NE MIB Statistics Policy (SNMP) dialog box appears.
- a. Select the MIB Entry Policies tab.
 - b. For each type of performance statistics you are collecting, enter its MIB entry name in the MIB Entry name field, and then press Enter (Return). Once you press Enter, the table of all routers that can be polled appears.

Statistic to Collect	MIB Entry Name
Interface performance	ifXEntry
Error packets in	ifEntry
Dropped packets out	
Memory available	sgiMemoryAvailable
Memory used	sgiMemoryUsed
Route Processor (CPU) utilization	tmnxSysCpuMonEntry

- c. From the table, select all routers that are to be polled.
- d. Click Properties.
 1. Change the Administrative State to Up.
 2. Set the Polling Interval to less than the interval at which the WAE Collector snapshot is run.

**Note**

Set the file policy Rollover and the accounting statistic Collection Interval to the same values. Refer to [Table 25-2](#) for recommended settings.

3. Click OK, and then click Yes to confirm you want to change the state. When you return to the MIB Entry Policies tab, notice the Administrative State column displays Up for each router to which you applied the change.
- e. Click Cancel to close the NE MIB Statistics Policy dialog box.
- f. Close the MIB Statistics Policy dialog box.

Configure SAM to Collect and Store Accounting Statistics

WAE Collector can import both LSP and interface queue statistics from SAM. As an alternative to LSP statistics, WAE Collector can import SDP statistics. All of these are a type of accounting statistic. To import these, you must first configure SAM by logging into the SAM server and creating two policies.

- [Create a File Policy](#) for storing the statistics.
- [Create an Accounting Policy](#) that collects the statistics and writes them to the file policy.

There must be a one-to-one mapping between each of these two policies. For example, if you create an accounting policy for collecting LSP statistics, then you would also create a file policy for storing those LSP statistics. The same is true for interface queue and SDP statistics.

After configuring SAM, you must then apply the accounting policies to the objects on the routers. See the [Apply Accounting Policies](#) section.

**Note**

There are combinations of Alcatel-Lucent router hardware, operating system, and SAM version that do collect LSP statistics. Contact your support representative for this information.

Create a File Policy

Create a file policy and distribute it to all routers for which the statistics will be collected ([Figure 25-3](#)).

-
- Step 1** Select the Tools->Statistics->File Policies menu. The File Policies dialog box appears.
- Step 2** Click Create. A File Policy, Global Policy dialog box appears.
- a. In the Displayed Name field, enter the file policy name.
 - b. Optional: Enter a description.
 - c. In the File area, Rollover (minutes) field, enter the interval at which the data file will be written over. By default, it is 1440, which means every 24 hours a new file is created.

**Note**

Set the file policy Rollover and the accounting statistic Collection Interval to the same values. Refer to [Table 25-2](#) for recommended settings.

- d. In the File area, Retention (hours) field, the recommendation is to keep the default of 12 hours. This value depends on the amount of storage capacity available.
- e. In the Drive list (Location area), select the primary location for storing the statistics.

- f. In the Storage Drive Backup list (Backup Location area), select the secondary location for storing the statistics in the event of a failure in the primary storage device.
 - g. Click Apply to accept the changes and close the File Policy dialog box.
 - h. Change the configuration mode from Draft to Released.
 1. In the Policy Configuration area, click Switch Mode to change the configuration mode from Draft to Released.
 2. Click Yes to confirm you want to change the mode.
 - i. Click Cancel to close the dialog box.
- Step 3** In the File Policies dialog box, select the newly created file policy.
- Step 4** Click Distribute to distribute the file policy to all the routers for which statistics will be collected. The Distribute-File dialog box appears.
- a. Select the routers to which you are distribute this policy. Use Ctrl-A (Cmd-A on Mac) to select all routers in the table.
 - b. Click the right arrow (->). The entries are moved from the left (Available Nodes) to the right (Selected Nodes).
 - c. Click Distribute.
 - d. Click Cancel to exit the Distribute-File dialog box.
- Step 5** Close the File Policies dialog box.

Figure 25-3 Example of Creating a File Policy for LSPs

Create an Accounting Policy

Create an accounting profile and distribute it to the routers for which statistics will be collected (Figure 25-4). Create only one accounting policy per type of traffic statistics you are collecting.



Note

L2/L3 means either L2 or L3.

- Step 1** Select the Tools->Statistics->Accounting Policies menu. The Accounting Policies dialog box appears.

Step 2 Click Create. An Accounting Policy, Global Policy dialog box appears.

- a. In the Displayed Name field, enter the accounting policy name.
- b. Optional: Enter a description.
- c. In the Type list, select the type of statistics to collect. Refer to [Table 25-2](#).

Since not all Alcatel-Lucent routers support the direct collection of LSP statistics, WAE Collector alternatively collects SDP traffic statistics. However, you can collect either LSP or SDP traffic statistics, but not both.

While WAE Collector uses only egress statistics, we support setting up the SAM server to collect a combination of network egress and ingress octets for interface queues.

Table 25-2 Accounting Statistics and Collection Interval

To Collect	Select Type	Set Collection Interval*
MPLS RSVP-TE LSP statistics	Combined MPLS LSP Egress	5
Interface queue statistics	Network Egress Octet	15
	Combined Network Ing Egr Octets	15
SDP statistics	Complete Service SDP Ingress Egress	5
L2/L3 VPN access interface statistics	Complete Service Ingress Egress	5

* Set the file policy Rollover and the accounting statistic Collection Interval to the same values. The `sam_getplan` tool uses the most recently gathered performance and accounting information gathered.

- d. Set the Collection Interval (m) field to the interval for collecting statistics. [Table 25-2](#) lists the recommended collection intervals.
- e. From the File area, click Select.
 1. Select the file policy that you want to associate with this accounting policy.
 2. Click OK.
- f. Click Apply to accept the changes.
- g. Change the configuration mode from Draft to Released.
 1. In the Policy Configuration area, click Switch Mode to change the configuration mode from Draft to Released.
 2. Click Yes to confirm you want to change the mode.
- h. Click Cancel to exit the dialog box.

Step 3 In the Accounting Policies dialog box, select the newly created accounting policy.

Step 4 Click Distribute to distribute the accounting policy to all the routers for which statistics will be collected. The Distribute-Accounting dialog box appears.

- a. Select the routers to which you are distributing this policy. Use Ctrl-A (Cmd on Mac) to select all routers in the table.
- b. Click the right arrow (->). The entries are moved from the left (Available Nodes) to the right (Selected Nodes).
- c. Click Distribute.
- d. Click Cancel to exit the Distribute-Accounting dialog box.

Step 5 Close the Accounting Policies dialog box.

Figure 25-4 Example of Creating an LSP Accounting Policy

The screenshot shows a 'Policy Configuration' dialog box with the following fields and options:

- Policy Scope:** Global Policy
- Configuration Mode:** Released (with a 'Switch Mode' button)
- Discovery State:** NIA (with an 'Origin: admin' field)
- ID:** 19
- Displayed Name:** Egress LSP Accounting
- Description:** NIA
- Type:** Combined MPLS LSP Egress (with a 'Default: false' dropdown)
- Collection Interval (m):** 5 (with a 'Use Default Interval: false' dropdown)
- File:** Name: LSP accounting policy, File ID: 29 (with 'Select...' and 'Properties' buttons)
- State:** Administrative: Up

Apply Accounting Policies

After the file and accounting policies have been created and distributed to the routers, you must apply the accounting policies to the objects on the routers. For example, if you created a file policy and accounting policy to collect and store LSP statistics, you must then apply that accounting policy to LSPs on the routers. Note that you can apply only one accounting policy of the same type.

After applying these accounting policies, you can verify that they have been applied using the SAM interface. See the [Verify Accounting Statistics Collection](#) section.

Following the steps in this section alleviates the need to manually configure the routers.


LSP Statistics

Follow these steps to associate an accounting policy with LSPs configured on the routers.

- Step 1** Select the Manage->MPLS->Dynamic LSPs menu. The Manage Dynamic LSPs dialog box appears.
- Step 2** Click Search to populate the list of dynamic LSPs.
- Step 3** Select all entries in the list by clicking on one and pressing Ctrl-A (Cmd-A on Mac).
- Step 4** Click Properties. The Dynamic LSP (Multiple Instances) dialog box appears.
 - a. Select the Accounting tab.
 - b. Click Select. The Select Accounting Policy dialog box appears.
 1. Select the newly created LSP accounting policy.
 2. Click OK to apply the accounting policy and close the dialog box.
 - c. In the Dynamic LSP (Multiple Instances) dialog box, click Apply and then click Yes to confirm the changes.
- Step 5** Close the Manage Dynamic LSPs dialog box.

Interface Queue Statistics

Follow these steps to associate an accounting policy with interface queues configured on the routers. A QoS network queue policy must be configured on these router ports.

-
- Step 1** Select the Manage->Equipment->Equipment menu. The Manage Equipment dialog box appears.
- Step 2** From the Select Object Type drop-down list, select
- For non-LAG members, select Port (Physical Equipment)->Physical Port (Physical Equipment).
 - For LAG members, select Port (Physical Equipment)->Logical Port (Physical Equipment).
- Step 3** Click Search to populate the list of ports.
- Step 4** Set the filter to “Mode EQUALS (Network) AND Equipped EQUALS (true).”
- a. Click the filter icon. The Manage Equipment - Filter dialog box appears.
 - b. Set the Mode value to Network.
 1. From the Attribute drop-down list, select Mode.
 2. From the Value drop-down list, select Network.
 3. Click Add.
 - c. Set the Equipped value to true.
 1. From the Attribute drop-down list, select States: Physical Port (Physical Equipment)->Equipped.
 2. From the Value drop-down list, select true.
 3. Click Add.
 - d. Click Apply.
 - e. Click Close to close the Manage Equipment - Filter dialog box.
- 
- Step 5** Select one of the ports, and then press Ctrl-A (Cmd-A on Mac) to select all ports.
- Step 6** Click Properties. A dialog box appears.
- a. Select the Policies tab.
 1. In the Network Queue area, click Select.
 2. Select the appropriate network queue policy, and click OK.
 - b. In the Accounting area, click Select.
 1. Select the accounting policy you want to apply, and click OK.
 2. Select the Collect Accounting Statistics option.
 3. Click Apply, and click Yes to confirm that you want to apply the accounting policy.
- Step 7** Click Cancel to close the Physical Port dialog box.
- Step 8** Close the Manage Equipment dialog box.

SDP Statistics

Follow these steps to associate an accounting policy with SDP configured on the routers.

-
- Step 1** Select the Manage->Service Tunnels menu. The Manage Service Tunnels dialog box appears.
- Step 2** Click Search to populate the list with SDPs (service tunnels).
- Step 3** Select one of the SDPs, and then press Ctrl-A (Cmd-A on Mac) to select all of them.

- Step 4** Select Properties. A Tunnel (Multiple Instances) dialog box appears.
- Select the Accounting tab.
 - Click Select. The Select Accounting Policy -Tunnel dialog box appears.
 - Select the accounting policy that you created for SDPs.
 - Click OK to close the dialog box.
- Step 5** Select the Collect Accounting Statistics option.
- Step 6** In the Tunnel (Multiple Instances) dialog box, click Apply and then click Yes to confirm you want to apply the accounting policy to the routers.
- Step 7** Click Cancel to close the Tunnel (Multiple Instances) dialog box.
- Step 8** Close the Manage Tunnel Services dialog box.

L2 or L3 VPN Access Interface Statistics

-
- Step 1** Select the Manage->Service->Services menu.
- Step 2** Click Search to populate the list with services. A Manage Services dialog box appears.
- Step 3** Select one of the VPN services in the list. For instance, these might be listed in the Service Name or Service ID columns. The service must have an associated L2/L3 access interface with an operational status of “up.”
- Step 4** Select Properties. A specific Service dialog box appears.
- Select the L2/L3 Access Interfaces tab.
 - Select all ports in the list.
 - Click Properties. The L2/L3 Access Interface - (Multiple Instances) dialog box appears.
 - Select the Accounting tab.
 - Click Select. The Select Accounting Policy dialog box appears.
 - Double-click the newly created VPN access interface.
 - Ensure the Collect Accounting Statistics check box is selected.
 - Click OK to close the dialog box.
- Step 5** Click OK in the Services dialog box, and click Yes to confirm.
- Step 6** Close the Manage Services dialog box.

Verify Accounting Statistics Collection

There are several ways to verify that you correctly configured the SAM server to collect statistics and that the accounting policies were pushed to the objects on the router.

- [Simple Verification](#) using the Tools menu
- [Per Object Verification](#) using the Manage menu
- [Router Configuration Verification](#)

Simple Verification



The simplest means of verifying the accounting policies is to select the Tools->Statistics->Accounting Retrieval Status menu. Find the accounting policy in the Last Policy ID Retrieved field, and check that “Success” is listed in the associated Stats Retrieval State column.

Per Object Verification

Step 1 The Manage menu you select depends on the object type you are verifying.

To Verify Accounting Policies For ...	Select ...
MPLS RSVP-TE LSP statistics	Manage->MPLS->Dynamic LSPs menu
Interface queue statistics <ul style="list-style-type: none"> • Non-LAG members • LAG members 	Manage->Equipment->Equipment menu <ul style="list-style-type: none"> • From the Select Object Type drop-down list, select Port (Physical Equipment)->Physical Port (Physical Equipment) • From the Select Object Type drop-down list, select Port (Physical Equipment)->Logical Port (Physical Equipment)
SDP statistics	Manage->Service Tunnels
L2/L3 VPN access interface statistics	Manage->Service->Services

Step 2 Click Search to populate the list area.

Step 3 Select all objects for which you need to verify that the accounting policy was applied.

Step 4 Click Properties.

Step 5 If you selected a service in step #3, follow these steps. Otherwise, go to step #6.

a. Select the L2/L3 Access Interface tab.

b. Select all ports on which you need to verify the statistics are collected.

Step 6 Select the Statistics tab.

Step 7 Select the object from the Select Object Type drop-down list.

To Verify this Object Type ...	Select ...
MPLS RSVP-TE LSP	Combined MPLS LSP Egress (Service Management)
Egress interface queue	Network Egress Octets (Service Management)

To Verify this Object Type ...	Select ...
Combined ingress and egress interface queue	Combined Network Egress Octets (Service Management)
SDP	Complete SDP Egress Packet Octets (Service Management)
L2/L3 VPN access interface	Complete Service Ingress Packet Octets (Service Management)
	Complete Service Egress Packet Octets (Service Management)

- Step 8** Verify the Time Captured column is populated with the time the statistics were last captured. Note for L2 or L3 access interface statistics, repeat this verification for both ingress and egress options in step #7.

Router Configuration Verification

Verify the configuration on the router using the `admin display-config` command. First verify that the file policy ID and accounting policy ID variables match those that you set. Next verify that the statistics are enabled (`no shutdown`). This section shows expected output for LSPs, interface queues, SDPs, and VPNs, as follows.

Object Type	Example Verification File Policy and Accounting Policies Were Set	Example Verification Statistics Are Enabled
MPLS RSVP-TE LSPs	Figure 25-5	Figure 25-6
Egress interface queues	Figure 25-7	Figure 25-8
Ingress and egress interface queues	Figure 25-9	Figure 25-10
SDPs	Figure 25-11	Figure 25-12
L2/L3 VPN access interfaces	Figure 25-13	Figure 25-14

Figure 25-5 Example Verification that LSP File Policy and Accounting Policy Are on the Router

```

log
  file-id 0
    description "Egress LSP stats"
    location cf3:
    rollover 15 retention 12
  exit
  accounting-policy 7
    description "Egress LSP statistics accounting"
    record combined-mpls-lsp-egress
    collection-interval 15
    to file 9
    no shutdown
  exit
exit

```

381437

Figure 25-6 Example Verification that Accounting Policy Was Applied to LSPs on the Router

```

router
  mpls
    lsp "timos01-timos02-00"
      to 192.168.31.8
      egress-statistics
        no shutdown
        collect-stats
        accounting-policy 7
      exit
    primary "p-timos01-timos02-00"
    exit
    no shutdown
  exit
exit

```

381438

Figure 25-7 Example Verification that Egress Interface Queue File Policy and Accounting Policy Are on the Router

```

log
  file-id 4
    description "Acme network"
    location cf2:
    rollover 15 retention 12
  exit

  accounting-policy 3
    description "Egress port statistics accounting"
    record network-egress-octets
    collection-interval 15
    to file 4
    no shutdown
  exit
exit

```

381435

Figure 25-8 Example Verification that Accounting Policy Was Applied to Egress Interface Queues on the Router

```

port 1/1/2
  ethernet
    encap-type dot1q
    mtu 1540
    network
      queue-policy "egress-queue-policy"
      accounting-policy 3
      collect-stats
    exit
  exit
  no shutdown
exit

```

381436

Figure 25-9 Example Verification that Combined Ingress and Egress Interface Queue File Policy and Accounting Policy Are on the Router

```

log

  file-id 9
    description "Combined network queue"
    location cf1: cf2:
    rollover 5 retention 12
  exit

  accounting-policy 1
    description "Combined Network QoS "
    record combined-network-ing-egr-octets
    to file 9
    no shutdown
  exit
exit

```

381433

Figure 25-10 Example Verification that Accounting Policy Was Applied to Combined Ingress and Egress Interface Queues on the Router

```

port 1/1/6
  ethernet
    network
      queue-policy "network-queue-policy"
      accounting-policy 1
      collect-stats
    exit
  exit
  no shutdown
exit

```

381434

Figure 25-11 Example Verification that SDP File Policy and Accounting Policy Are on the Router

```

log
  file-id 66
    description "SDP statistics"
    location cf3:
    rollover 5 retention 12
  exit

  accounting-policy 66
    description "SDP accounting policy "
    record complete-sdp-ingress-egress
    to file 66
    no shutdown
  exit
exit

```

381439

Figure 25-12 Example Verification that Accounting Policy Was Applied to SDPs on the Router

```

service

  sdp 1 mpls create
    far-end 192.168.31.8
    lsp "timos01-timos02-00"
    path-mtu 9194
    keep-alive
    no shutdown
  exit
  collect-stats
  accounting-policy 66
  no shutdown
exit
  epipe 10 customer 10 vpn 10 create
    service-mtu 1518
    sap 1/1/2 create
      ethernet
      llf
    exit
  exit
  spoke-sdp 1:10 create
    collect-stats
    accounting-policy 66
  exit
  no shutdown
exit
exit

```

381440

Figure 25-13 Example Verification that VPN File Policy and Accounting Policy Are on the Router

```

log
  file-id 7
    description "Complete Service Ingress Egress"
    location cf1:
    rollover 5 retention 12
  exit

  accounting-policy 7
    description "Complete Service Ingress Egress"
    record complete-service-ingress-egress
    to file 7
    no shutdown
  exit
exit

```

381441

Figure 25-14 Example Verification that Accounting Policy Was Applied to VPNs on the Router

```

service

  customer 10 create
    description "Acme voice VPN customer 10"
  exit
  epipe 10 customer 10 vpn 10 create
    service-mtu 1518
    sap 1/1/2 create
      description "to-ftp-client"
      ethernet
        11f
      exit
      collect-stats
      accounting-policy 7
    exit
    spoke-sdp 1:10 create
    exit
  no shutdown
exit
exit

```

381442

SAM Integration

The SAM Get Plan (`sam_getplan`) tool generates plan files through interfacing with 5620 SAM servers. Generated plan files include network topology and traffic measurements. SAM Get Plan implements options needed to retrieve network topology and traffic measurements from a network managed by a SAM server.

- Establishes a session with the SAM server.
- Discovers network topology from OSPF or IS-IS parameters.
- Discovers router model and SR OS version.

- Optional
 - Discovers MPLS RSVP-TE LSPs, interface queues, LAG ports (members), SRLG members, and VLL Epipe, VPLS, and VPRN VPNs.
 - Collects traffic statistics for interfaces, interface queues, LAG ports, and either RSVP-TE LSPs or SDPs.
- The `sam_getplan` tool creates a plan file. The SAM Get Plan GUI tool opens the plan file automatically, which you must then save to store it.

sam_getplan

The `sam_getplan` integration tool interfaces with SAM servers to provide simulation and traffic engineering for SAM-managed networks. A single call to `sam_getplan` retrieves the network topology, as well as current or historical interface traffic measurements, which can then be used as input to Demand Deduction. The tool queries each router in the discovered network to obtain additional information, such as router name, vendor, and model. It also creates a plan file. For more information, see the `sam_getplan` Help. For information on using this tool from the WAE Design GUI, see the *WAE Design Integration and Development Guide*.

Example: This is a basic example of collecting OSPF topology and interface traffic.

```
sam_getplan -out-file ~/us-wan.txt -server sam.vendorisp.com -user samuser -pass
2021385t12152251911114a -util-stats loggedstats
```

Example: This is an example of gathering LSP actual path and traffic measurements. After executing it, run the `resolve_plan` tool with its default options on the newly generated plan file to resolve destination nodes in unrouted LSPs.

```
sam_getplan -out-file isp2.txt -server sam.isp2.com -user isp2_user -pass isp2_pass
-util-stats loggedstats -include-lsp-measurements true -lsp-actual-path actual -log-file
isp2-sam_getplan.log
```

Example: This is an example of VLL VPN topology discovery.

```
sam_getplan -out-file vpn_test.txt -server 172.16.0.219 -user isp4_admin -pass isp4_pass
-server-protocol https -port 8443 -vpn-type VLL:Epipe
```

The SAM version used by `sam_getplan` is listed in the Comments column of the NetIntHistory table.



Note

The `sam_getplan` intervals interact with the SAM server configurations. The file policy Rollover and accounting statistic Collection Interval must be set to the same value on the SAM server. If needed, make adjustments using the `sam_getplan -num-logged-traff-level` option, `-logged-measwin-length` option, or both. The `-logged-measwin-length` option must be longer than the maximum Collection Interval.

Snapshot Integration

The result of the SAM server discovery can be fed into the snapshot process that collects network data using other WAE Collector CLI tools. The resulting plan file can be inserted into an archive for use with the applications. This integration enables you to more easily use WAE Collector in environments containing both Alcatel and non-Alcatel networks.

The `snapshot.txt` file includes a `<SAM_GETPLAN>` task, which must be executed before any other snapshot tasks. The `snapshot.txt` file also contains sam environment variables that are used only by the `sam_getplan` tool. No other tools in the `snapshot.inc` file use these SAM variables. There is an accompanying default `sam_getplan` defined in the `snapshot.inc` file.

The following table shows the sequence that `snapshot.txt` tasks must follow. As with other collections, you can enable other tasks. For more information on configuring `snapshot.txt` and `snapshot.inc` files, see the [Snapshot Files](#) chapter.

```
SAM_GETPLAN
LOGIN_FIND_IGP_DB
SNMP_FIND_NODES
SNMP_FIND_INTERFACES
SNMP_FIND_RSVP (optional)
SNMP_FIND_VPN (optional)
SNMP_POLL
```

Discovering Multi-Vendor Networks with SAM

If you are discovering a network containing both Alcatel and non-Alcatel nodes, you must configure the `net_access.txt` file to tell the online tools to ignore the Alcatel nodes for specific objects. For more information about configuring this file, see the [Network Access File](#) chapter.

Example: This `net_access.txt` example tells `snmp_find_rsvp` to ignore Alcatel nodes when discovering LSPs and collecting their statistics, and tells `snmp_poll` not to collect any statistics from Alcatel nodes.

<PerRouterSettings>				
NodeRegExp	IPRegExp	TaskRegExp	RouterMode	SQLFilter
		snmp_find_rsvp	Ignore	Vendor='Alcatel'
		snmp_poll	Ignore	Vendor='Alcatel'

Related Topics

- [Snapshot Files](#) chapter
- [Manual Collection](#) chapter
- *WAE Design Plan Table Schema and CLI Reference*
- *WAE Design Integration and Development Guide*



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 $CARIDEN_ROOT/etc/auth-acme.enc` (where `$CARIDEN_ROOT` is the installation directory)

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—User name 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 the [Add Router-Specific Authentication Information](#) section.

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 26-1](#)).
- <UserTable>—Contains usernames and passwords for login access to nodes ([Table 26-2](#)).
- <CommunityTable>—Contains SNMPv2c community strings for access to nodes ([Table 26-3](#)).
- <SNMPv3ProfileTable>—Contains SNMPv3 profiles, which define a set of authentication, encryption, and context information ([Table 26-4](#)).
- <SNMPv3MappingTable>—Defines how to match routers with the SNMPv3 profiles ([Table 26-5](#)).



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 26-1 **<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 26-2 **<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 26-3 *<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 26-4 *<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 26-5 *<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 the [Tables in the Authentication File](#) section). 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 further 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.

Related Topics

- [Snapshot Files](#) chapter
- [Manual Collection](#) chapter
- [Continuous Polling and PCEP LSP Collection](#) chapter



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 used in the last collection by the Collector server. For information on configuring the Collector server, see the [WAE Collector UI Overview](#) chapter.

Best Practices

- Make a copy of the default `net_access.txt` file located in `$CARIDEN_HOME/etc` before editing it, and put the edited version in `$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 27-1](#) 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 27-2](#) provides an example <GlobalSettings> table. The empty field at the beginning of the last row means *everything except* `snmp_poll` and `snmp_find_*`.

Table 27-1 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 27-2 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 27-3](#) 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 27-4](#) 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 27-3 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

Name	Property	Value
Normal	SNMP_RSVP_stats_method	Default
Junos_new	SNMP_RSVP_stats_method	Method1
Junos_old	SNMP_RSVP_stats_method	Method2

Table 27-4 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_rsvpl|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.

Related Topics

- [Snapshot Files](#) chapter
- [Manual Collection](#) chapter
- [Continuous Polling and PCEP LSP Collection](#) chapter



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](#)
- [Update Summary of Time-Sequence Plot Data](#)
- [Insert or Extract Files from an Archive](#)
- [Manage Archives for WAE Design Archive](#)
- [Make Batch Changes to Archive Files](#)

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 *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 *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 selecting the File->Save to->Design Archive to menu in the WAE Design GUI. For information, see the *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 selecting the File->Open from->Design Archive or File->Open from->WAE Live menu in the WAE Design GUI. For information, see the *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 are not applicable 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 28-1](#) 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 28-1 Valid Variables for the `-cmd` Argument of `archive_do`

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

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"
```

Related Topics

- [Snapshot Files](#) chapter
- *WAE Design Archive User and Administration Guide*
- *WAE Live Configuration Guide*



WAE Design GUI and Remote WAE Core Server

From the WAE Design GUI, you can extract .pln plan files from a remote WAE Core server¹, make whatever modifications are necessary, and re-insert them to the remote server. It is important to note that these plan files are not stored with the plan files generated by WAE Collector.

The Network Modeler Module (on the WAE Core server) contains the *working plan file*, which represents the current state of the network. It can also contain one or more *staging areas*, where each staging area enables you to work on plan files before deploying them.

Saving the plan file to the working area of the Network Modeler Module overwrites the existing plan file. This means that any newly discovered or recently modified plan files are overwritten. For this reason, if you are not deploying the plan file, it is a best practice to access a staging area.

Location

Server: 293.268.34.234

Protocol: HTTPS

Port: 7777

Username: ObiDusan

Password:

Save Settings

Save Password

Stage ID: 77

Operation: Load Plan, Load Plan, Process Plan, Deploy Plan

380656



Note

It is strongly recommended that you verify a plan file before deploying it to the network.

Prerequisites

- If using the staging area, it must already exist on the Network Modeler Module, and you must know the stage ID.
- The `$CARIDEN_ROOT/software/wae-core/etc/com.cisco.wano.nsps.nbrs.cfg` file must either have the `authenticationEnabled` property set to `false`, or you must know the values entered as the username and password properties.

Step 1

Use the File menu to start the process.

- To open a plan file from the WAE Core server, select the File->Open from->WAE menu.
- To save a plan file to the WAE Core server or to deploy a plan file to the network, select the File->Save to->WAE menu.

1. The Network Modeler Module, Optimization and Prediction Module, Calendaring Module, and Deployer Module all use the WAE Core server.

- Step 2** After opening the dialog box, enter the hostname or IP address of the server.
- Step 3** Identify how to connect to the server by selecting the appropriate protocol and entering the port number (for example, HTTP 7777).
- Step 4** If the authentication is enabled, enter the same username and password.
- Step 5** Optional: Save the data you entered for future use.
- To save all settings except the password, select Save Settings.
 - To save the password for future use, select Save Password.
- Step 6** If using a staging area for this access, enter the stage ID number.
- Step 7** If saving a plan file, select the operation, as follows.
- Load Plan—Save a plan file to the WAE Core server. No further processing is done to this plan file on the platform.
 - Process Plan—Save a plan file to the WAE Core server and simulate failure scenarios such that upon querying a demand, worst-case analysis is automatically performed on interfaces, circuits, and SRLGs.
 - Deploy Plan—Compare the plan file with the working plan file that currently resides in the Network Modeler Module, and then deploy the differences (LSPs only) directly to the network. A copy of it is also saved to the working plan file area in the Network Modeler Module.
- Step 8** Click OK.