Supported Collectors and Tools

- Collector descriptions, on page 1
- Run an external script as a startup script, on page 3
- Collecting basic topology information, on page 5
- Collect LSP information, on page 11
- Collect PCEP LSP information using SR-PCE, on page 13
- Collect multicast flow data from a network, on page 14
- Discover BGP peers, on page 17
- Discover VPN topology, on page 19
- Inventory collector and hardware tables, on page 21
- Collect port, LSP, SRLG, and VPN information using configuration parsing, on page 27
- Collect Circuit Style RSVP-TE information, on page 30
- Configure the Layout collector for improved network model visualization, on page 32
- Collect traffic statistics, on page 33
- Collect traffic demands information, on page 37
- NetFlow data collection, on page 39
- Run an external script against a network model, on page 42
- How data is collected from third-party devices, on page 45
- Merge AS plan files, on page 47
- Representative plan files, on page 48

Collector descriptions

Each collector in Cisco Crosswork Planning has capabilities that determine what it collects or deploys. This table summarizes the collectors and their functions.

Table 1: Collector descriptions

Collector	Description	Prerequisites and notes	Configuration steps
Basic Topology Collection			
IGP database	Discovers IGP topology using login and SNMP.	This is a basic topology collection. The resulting network model is used as the source network for other collectors.	See Collect topology information using the IGP database collector, on page 6

Collector	Description	Prerequisites and notes	Configuration steps
SR-PCE	• Discovers Layer 3 topology using SR-PCE. • Uses raw SR-PCE data as the source for the topology. • Discovers node, interface, and port properties using SNMP.	• Configure SR-PCE agents before running this collection. For details, see Configure agents. • This is a basic topology collection for networks using SR-PCE. The resulting network model is used as the source network for other collectors.	See Configure the SR-PCE collector to collect topology information, on page 8
Advanced Modeling Collection			
LSP	Discovers LSP information using SNMP.	• A network model with basic topology collection must exist. • If using SR-PCE, collect the topology information using the SR-PCE collector, before running this collection. For details, see Configure the SR-PCE collector to collect topology information, on page 8.	See Collect LSP information, on page 11
PCEP LSP	Discovers PCEP LSPs using SR-PCE. **Note** This collector is accessible only when SR-PCE collector is selected as the basic topology collector.	Collect the topology information using the SR-PCE collector before running this collection. For details, see Configure the SR-PCE collector to collect topology information, on page 8.	See Collect PCEP LSP information using SR-PCE, on page 13
BGP	Discovers BGP peering using login and SNMP.	A network model with basic topology collection must exist.	See Discover BGP peers, on page 17
VPN	Discovers Layer 2 and Layer 3 VPN topology.	A network model with basic topology collection must exist.	See Discover VPN topology, on page 19
Config parsing	Discovers and parses information from router configurations in the network.	A network model with basic topology collection must exist.	See Collect port, LSP, SRLG, and VPN information using configuration parsing, on page 27
Traffic and Demands Collection			
Inventory	Collects hardware inventory information.	A network model with basic topology collection must exist.	See Collect hardware inventory information
Multicast	Collects multicast flow data from a given network.	A network model with basic topology collection must exist.	See Collect multicast flow data from a network, on page 14

Collector	Description	Prerequisites and notes	Configuration steps
Layout	Adds layout properties to a source model to improve visualization.	• An aggregated network model. • After you configure the Layout collector, import a plan file containing layout properties into the Layout model.	See Configure the Layout collector for improved network model visualization, on page 32
Traffic collection	Collects traffic statistics (interface traffic, LSP traffic, MAC traffic, and VPN traffic) using SNMP polling.	• A network model with basic topology collection must exist. • If collecting LSP traffic, a network model with LSP collection must exist. See Collect LSP information, on page 11. • If collecting VPN traffic, a network model with VPN collection must exist. See Discover VPN topology, on page 19.	See Collect traffic statistics, on page 33
Demand deduction	Collects information regarding traffic demands from the network.	Source DARE network containing traffic data must exist.	See Collect traffic demands information, on page 37
NetFlow	Collects and aggregates exported NetFlow and related flow measurements.	A network model with basic topology collection must exist.	See Configure the NetFlow collection, on page 40
Custom Scripts			
External script	Runs customized scripts to append additional data to a source network model.	A source network model and a custom script must exist.	See Run an external script against a network model, on page 42

Run an external script as a startup script

This topic describes how to run an external script as a first step in a collection configuration chain.

You can provide an external script as the initial step in a data collection chain. When enabled, the startup script is executed before any other collectors in the chain. This feature allows greater flexibility in how data is gathered and processed during collection.

If a startup script is used as the first step, the IGP database or SR-PCE collector becomes optional. Its configuration section will have a Source drop-down list enabled. This source is not used by the basic topology collectors for data collection. It is used to determine the order of execution of these collectors after the startup script and other external scripts in basic topology.

Before you begin

- Complete the steps mentioned in Preconfiguration workflow.
- Review Important notes on custom startup scripts, on page 5.
- Have the custom script and any supporting files ready in one of the accepted file formats or compressed archives.

Procedure

Step 1 Decide whether to create a new collection or edit an existing one. For details, see Configure a collection or Edit a collection.

Step 2 In the **Startup script** section, select **Script**.

Step 3 (Optional) If you choose a startup script, you may skip the Basic topology collector, or configure it with the startup script as its source. Select any additional collectors as needed.

Step 4 On the Configure page, enter the script details. The startup script configuration is similar to that of other external scripts except that it does not require a source.

Option	Description
Collector name	Specify the name for the collection.
Is source a plan file?	Check this check box if you want to run the script on a plan file. If you select this option, enter the plan file details in the **Input plan file** field.
Input file	Upload your custom script and any supporting files necessary for its successful execution. If multiple files are required, compress them into a single archive before uploading. Valid file formats are .py, .sh, .pl, .zip, .tar, .gz, and .tar.gz. **Note** Each time a file is uploaded, the input file option is overwritten.
Executable script	Enter the name of the file that initiates the script execution process. This is one of the files you uploaded in the **Input file** field. For more information, see Run an external script against a network model, on page 42.
Script language	Select the language of the custom script. The valid script languages are Python, Shell, and Perl.
Aggregator properties	If you want to specify any tables or columns to be aggregated, then list them in a .properties file and upload the file using this field. By default, all columns and tables are aggregated.
Timeout	Specify the action timeout. The default is 30 minutes.

Step 5 (Optional) If you selected other collectors in Step 3, configure their parameters as needed.

To use the startup script as a source for any of the collectors, while configuring the collector parameters, select the startup script name from the **Source** drop-down list.

Step 6 Click **Next**.

Step 7 Preview the configuration and then click **Create** to create the collection.

Step 8 Configure schedules for the collection job. You can schedule the collection job to run immediately or at specific intervals. For details, see Schedule a collection.

Your custom script executes as a first step in the collection configuration chain.

Important notes on custom startup scripts

Review these points when using custom startup scripts in your collections:

- Only one startup script is allowed per collection chain.
- The aggregation of any database file produced by the startup script depends on the aggregator properties set in the collector configuration.
- If you configure a collector to use a startup script as its source and the script does not produce a valid database file, the collector execution will fail.
- When migrating or restoring configurations from earlier releases, ensure that all required startup script options are available and properly populated.
- If you are using a script from Cisco WAE and intend to use it within Cisco Crosswork Planning, it may not function as expected without modifications. This is due to architectural differences between Cisco WAE and Cisco Crosswork Planning, including variations in how the files are referenced. You must adjust the script appropriately for use in Cisco Crosswork Planning.

Collecting basic topology information

Summary

Collecting basic topology information involves selecting and configuring a basic topology collector to build an initial network model that serves as the source for further data collections in Cisco Crosswork Planning. Selecting the appropriate topology collector determines which data sources are included.

Two key collectors involved in the process are:

- **IGP database**: Discovers IGP topology using login and SNMP.
- **SR-PCE**: Discovers Layer 3 topology using BGP-LS via SR-PCE.

You can select only one collector per collection to gather topology information. Selecting both collectors at the same time is not allowed.

Workflow

These are the stages of collecting basic topology information.

1. Select either the **IGP database** or the **SR-PCE** collector to gather topology information for a given collection.
2. Configure the chosen collector based on your requirements.
3. Generate a network model from the collected data, which becomes the source for additional data collections.

For step-by-step instructions on configuring the **IGP database** and **SR-PCE** collectors, see Collect topology information using the IGP database collector, on page 6 and Configure the SR-PCE collector to collect topology information, on page 8.

Collect topology information using the IGP database collector

This topic describes how to configure the **IGP database** collector to discover complete network topology using IGP database.

The **IGP database** collector discovers network topology by leveraging the IGP database for node properties and SNMP for interface and port discovery. It is typically the first collector you configure because it provides the foundational network data required by other collectors. It supports multiple OSPF and IS-IS instances. All links collected from routers will have an associated IGP process ID. The resulting network model is used as the source network for additional collections, because it provides the core node, circuit, and interface information used by other collectors.

Follow these steps to collect topology information using the IGP database collector.

Before you begin

- Complete the steps mentioned in Preconfiguration workflow.
- Ensure you have network credentials and access for routers to be used as seed routers.

Procedure

Step 1 Decide whether to create a new collection or edit an existing one. For details, see Configure a collection or Edit a collection.

Step 2 In the **Basic topology** section, select **IGP database** and click **Next**.

Step 3 On the Configure page, under **Seed router**, enter these configuration parameters:

- **Index**: Enter a unique index number for the seed router.
- **Router IP**: Enter the management IP address for the seed router.
- **Protocol type**: Select the IGP protocol running on the network. The options are: ospf, ospfv3, isis, and isisv6.

If you select ...	Then ...
ospf or **ospfv3**	Enter the value for **OSPF area** on the **Advanced** page (click ⚙). The OSPF area option specifies the area ID or all. The default is area 0.
isis or **isisv6**	Enter the value for **ISIS level** (1, 2, or BOTH) on the **Advanced** page (click ⚙). The default is level 2.

- **Collect interfaces**: Ensure this box this checked to discover the full network topology. This option is enabled by default.

Step 4 (Optional) To add more seed routers, click **+ Add router** and repeat Step 3 for each seed router. Assign a unique index number to every seed router.

Step 5 (Optional) To include or exclude specific QoS node information, expand **Advanced settings** > **QoS Node Filter**, then click **+ Add node filter** and enter the required values.

Step 6 (Optional) Expand the **Advanced settings** panel and configure any other relevant advanced fields as needed. For descriptions of these advanced options, see IGP and SR-PCE collection advanced options, on page 9.

Step 7 Click **Next**.

Step 8 Preview the configuration and then click **Create** to create the collection.

Step 9 Configure schedules for the collection job. You can schedule the collection job to run immediately or at specific intervals. For details, see Schedule a collection.

The IGP database collector begins the topology discovery process, building a network model using the specified seed routers and advanced configuration options.

What to do next

Use this collection as the source network to configure additional collections. For details on configuring different collectors, see the relevant topics in this chapter. For details on editing the collection, see Edit a collection.

SR-PCE agent and collector

The SR-PCE agent and SR-PCE collector are Cisco Crosswork Planning components that enable communication and telemetry data collection between SR-PCE servers and the network.

SR-PCE agent

An SR-PCE agent is a Cisco Crosswork Planning component that

- connects to the SR-PCE server and processes the telemetry data sent by the server
- uses two different REST connections with SR-PCE: one for LSP data collection and another for topology data collection, and
- can optionally subscribe to SR-PCE and listen to further network change events after collecting topology and LSP data.

SR-PCE collector

An SR-PCE collector is a Cisco Crosswork Planning component that

- captures network updates for any changes in IGP Metric, Delay, and Node Overload
- populates the FlexAlgoAffinities, FlexAlgorithms, SRv6NodeSIDs, SRv6InterfaceSIDs, NodePrefixLoopbacks, and NodeSIDPrefixLoopbacks tables, and
- reads the LocalDomainIdentifier column of NetIntXtcLinks and populates the IGP Process ID in the Interfaces table.

The SR-PCE collector does not populate the SRv6NodeSIDPrefixLoopbacks table because the loopback address associated with SRv6 is not obtained using SR-PCE. To populate the SRv6NodeSIDPrefixLoopbacks details, add an external script while configuring the collector. Otherwise, the cross-table filter from SRv6NodeSIDs to NodePrefixLoopbacks will not display any results in the Cisco Crosswork Planning Design application. For details on running the external scripts, see Run an external script against a network model, on page 42.

Methods for topology discovery

The network model resulting from topology discovery is used as the source network for additional collections. It provides the core node, circuit, and interface information used by other collectors.

Topology and interface or port properties can be discovered in two ways.

- Using SNMP: Preferred for network discovery, as it retrieves detailed node and interface or port properties.
- Using SR-PCE only (the Extended discovery field disabled): Useful for testing, or if SNMP is unavailable.

Important notes on SR-PCE topology collection

- The default ISIS level is set to level 2 for NodePrefixLoopbacks. The OSPF network uses the same value.
- Cisco Crosswork Planning does not reflect changes from a non-null value to a null value in the FlexAlgo columns. The updated values start reflecting after a DARE re-sync.
- During data collection, dual stack support (the ability to handle both IPv4 and IPv6 simultaneously) and configuration of OSPF or ISIS on an interface are populated correctly. However, if both OSPF and ISIS are enabled on a single interface for data collection, dual stack and its interface resolution are not supported during SR-PCE collection.
- The IPv4 metric value is populated in the IGP metric table and the IPv6 value is populated in the IPv6-IGP metric table. The TE metric values are updated in the same way.
- The SR-PCE collector can collect Application-Specific Link Attribute (ASLA) delay information for interfaces and typically updates this information in the database in real time. However, if the collector receives several consecutive topology update events from SR-PCE within one minute, it may record the changes only during the next collection. In rare cases, the update may take effect only after manually restarting the SR-PCE agent.

Configure the SR-PCE collector to collect topology information

This topic describes how to configure the **SR-PCE** collector to collect Layer 3 topology information using SR-PCE.

Follow these steps to configure the SR-PCE collector.

Before you begin

- Complete the steps mentioned in Preconfiguration workflow.
- Ensure an SR-PCE agent is configured and running. For details on agent setup, see Configure agents.

Procedure

Step 1 Decide whether to create a new collection or edit an existing one. For details, see Configure a collection or Edit a collection.

Step 2 In the **Basic topology** section, select **SR-PCE** and click **Next**.

Step 3 On the Configure page, enter these configuration parameters:

- **SR-PCE host**: Select an SR-PCE agent.

- **Backup SR-PCE host**: Select a backup SR-PCE agent. If you do not have a backup, leave this field empty. Ensure you do not use the same SR-PCE agent as both the **SR-PCE host** and the **Backup SR-PCE host**.
- **ASN**: Enter 0 to collect information from all autonomous systems in the network, or enter the autonomous system number (ASN) to collect information only from a particular ASN. For example, if the SR-PCE agent has visibility to ASN 64010 and ASN 64020, enter 64020 to collect information only from ASN 64020.
- **IGP protocol**: Select the IGP protocol that is running on the network.
- **Extend discovery**: Check the **Enabled** check box to discover the full network topology (nodes and interfaces).
- **Reactive network**: Check the **Enabled** check box to subscribe to notifications from SR-PCE to update the addition or deletion of nodes or links.
- **Trigger collection**: Check the **Enabled** check box to collect topology collection on new topology additions (nodes or links).

Step 4 (Optional) Expand the **Advanced settings** panel and configure any other relevant advanced fields as needed. For descriptions of these advanced options, see IGP and SR-PCE collection advanced options, on page 9.

Step 5 Click **Next** to continue.

Step 6 Preview the configuration and then click **Create** to create the collection.

Step 7 Configure schedules for the collection job. You can schedule the collection job to run immediately or at specific intervals. For details, see Schedule a collection.

The SR-PCE collector initiates topology discovery, gathers Layer 3 topology information, and updates the network model with the collected data.

What to do next

Use this collection as the source network to configure additional collections. For details on configuring different collectors, see the relevant topics in this chapter. For details on editing the collection, see Edit a collection.

IGP and SR-PCE collection advanced options

You can configure several advanced options when using the IGP database and SR-PCE collectors.

Table 2: IGP and SR-PCE collection advanced options

Option	Description
Options applicable for both IGP and SR-PCE collection:	
Nodes	
Node performance collection	Collects node performance data if enabled.
Remove node suffix	Removes node suffixes from node names if the node contains the specified suffix. For example, 'company.net' removes the domain name for the network.
QoS queues	Allows interfaces (configured with QoS in the router) to display QoS information.

Option	Description
Data collection timeout	Sets the maximum time allowed for data collection, in minutes. If the specified limit is exceeded, the internal tools used for data collection will time out and exit. The default is 60 minutes.
QoS node filter	Defines a filter to determine the nodes for which the QoS data is collected.
Interfaces	
Find parallel links	Finds parallel links that are not in the IGP database when IS-IS TE extensions are not enabled.
IP guessing	Indicates the level of IP address guessing to perform for interfaces that are not present in the topology database. This setting is used when IS-IS TE extensions are not enabled. • OFF: Performs no guessing. • Safe: Makes guesses only when there is no ambiguity. • FULL: Makes best-guess decisions when there is ambiguity.
Port LAG discovery	Enables LAG discovery of port members.
LAG port match	Determines how to match local and remote ports in port circuits. • Guess: Creates port circuits to match as many ports as possible. • Exact: Matches based on LACP. • Complete: Matches based on LACP first, and then tries to match as many as possible. • None: Does not create port circuits.
Cleanup circuits	Removes circuits that do not have IP addresses associated with interfaces. Circuit removal is sometimes required when there are IS-IS advertising inconsistencies in the IS-IS database.
Copy description	Copies physical interface descriptions to logical interfaces if there is only one logical interface and its description is blank.
Physical ports	Collects L3 physical ports for Cisco devices.
Minimum IP guessing	Specifies the minimum prefix length for IP guessing. All interfaces with equal or larger prefix lengths are considered.
Minimum prefix length	Specifies the minimum prefix length allowed when finding parallel links. All interfaces with equal or larger prefix lengths (but less than 32) are considered.
Data collection timeout	Sets the maximum time allowed for data collection, in minutes. If the specified limit is exceeded, the internal tools used for data collection will time out and exit. The default is 60 minutes.
Debug	

Option	Description
Verbosity	Sets the level of detail in log messages. The default is 30, with a valid range from 1 to 60.
Net recorder	Records SNMP messages. The options are Off, Record, and Playback. The default is Off. • Record: The SNMP messages to and from the live network are recorded internally as discovery runs. It is used for debugging. • Playback: The recorded messages are played back through the collector as if they came from the live network, thus providing offline debugging of network collection. • Off: No recording or playback is performed.
Option applicable only for SR-PCE collection:	
Single-ended eBGP discovery	Discovers eBGP links that have only a single link end. This scenario is not common.

Collect LSP information

This topic describes how to configure the **LSP** collector to collect the RSVP LSP information in the network using SNMP.

Before you begin

Complete the steps mentioned in Preconfiguration workflow.

Procedure

Step 1 Decide whether to create a new collection or edit an existing one. For details, see Configure a collection or Edit a collection.

Step 2 To use an external script as a first step of the collection configuration chain, select the startup script option. When using a startup script, you can either skip the basic topology collector or configure it to use the startup script as its source. If you do not use a startup script, you must select one of the basic topology collectors as needed.

Step 3 In the **Advanced modeling** section, select **LSP** and click **Next**.

Step 4 On the Configure page, click **LSP** in the **Selected collectors** pane on the left.

Note
Ensure that the basic topology parameters are updated to meet your needs. Update the parameters if necessary.

Step 5 Enter these configuration parameters:

- **Source**: Select the source collector whose output serves as the input for this collector.
- **Get FRR LSPs**: Check the **Enabled** check box to discover MPLS Fast Reroute (FRR) LSP (backup and bypass) information.

Step 6 (Optional) Expand the **Advanced settings** panel and enter the details in the relevant fields. For descriptions of these advanced options, see LSP collection advanced options, on page 12.

Step 7 Click **Next**.

Step 8 Preview the configuration and then click **Create** to create the collection.

Step 9 Configure schedules for the collection job. You can schedule the collection job to run immediately or at specific intervals. For details, see Schedule a collection.

The LSP collector is set up and scheduled according to your configuration.

What to do next

Use this collection as the source network to configure additional collections. For details on configuring different collectors, see the relevant topics in this chapter. For details on editing the collection, see Edit a collection.

LSP collection advanced options

You can configure several advanced options when using the LSP collector.

Table 3: LSP collection advanced options

Option	Description
Use calculated hops	Uses the calculated path hops table instead of the actual path hops table when discovering path hops.
Find actual path	Discovers actual paths for LSPs.
Get extras	Collects additional LSP properties.
Use signaled name	Uses the LSP tunnel signaled name instead of the LSP tunnel name (IOS-XR). **Note** To retrieve the signaled name when using Config parsing with the LSP collector, ensure the LSP collector executes before the Config parsing collector. If you do not follow this order, the LSP tunnel name collected by Config parsing will overwrite the signaled name value collected by the LSP collector.
Auto bandwidth	Discovers auto bandwidth.
Data collection timeout	Sets the maximum time allowed for data collection, in minutes. If the specified limit is exceeded, the internal tools used for data collection will time out and exit. The default is 60 minutes.
Debug	
Verbosity	Sets the level of detail in log messages. The default is 30, with a valid range from 1 to 60.

Option	Description
Net recorder	Records SNMP messages. The options are Off, Record, and Playback. The default is Off. • Record: The SNMP messages to and from the live network are recorded internally as discovery runs. It is used for debugging. • Playback: The recorded messages are played back through the collector as if they came from the live network, thus providing offline debugging of network collection. • Off: No recording or playback is performed.

Collect PCEP LSP information using SR-PCE

This topic describes how to configure the **PCEP LSP** collector.

The PCEP LSP collector uses the data collected from the SR-PCE collector and appends LSP information, allowing you to generate a new and enhanced network model.

Before you begin

- Complete the steps mentioned in Preconfiguration workflow.
- Complete the BGP-LS topology collection for your network using the SR-PCE collector. You need to use this model as the source network for collecting LSP information. For more information, see Configure the SR-PCE collector to collect topology information, on page 8.

Procedure

Step 1 Decide whether to create a new collection or edit an existing one. For details, see Configure a collection or Edit a collection.

Step 2 In the **Basic topology** section, select **SR-PCE**.

Step 3 In the **Advanced modeling** section, select **PCEP LSP** and click **Next**.

Step 4 On the Configure page, click **PCEP LSP** in the **Selected collectors** pane on the left.

Note
Ensure that the basic topology parameters are updated to meet your needs. Update the parameters if necessary.

Step 5 Enter these configuration parameters:

- **Source**: Select the source collector whose output serves as the input for this collector.
- **Agents**: Select the SR-PCE agents from the drop-down list. For information on creating agents, see Configure agents.

  Note
  When using multiple SR-PCE agents, note that each additional agent may increase the overall execution time, depending on the data volume the collector has to process for each agent. Consider this aspect to ensure optimal performance when selecting multiple agents.

- **Reactive network**: Check the **Enabled** check box to subscribe to notifications from SR-PCE for real-time LSP updates. This option is enabled by default.

Step 6 (Optional) Expand the **Advanced settings** panel and enter these information:

- **RSVP use signaled name**: Check the **Enabled** check box to use the RSVP LSP tunnel signaled-name instead of LSP tunnel name (IOS-XR).
- **SR use signaled name**: Check the **Enabled** check box to use the SR LSP tunnel signaled-name instead of LSP tunnel name (IOS-XR).
- **SR add index**: Check the **Enabled** check box to add indexes to SR LSP tunnels from associated interfaces (IOS-XR).
- **Data collection timeout**: Set the maximum time allowed for data collection, in minutes. If the specified limit is exceeded, the internal tools used for data collection will time out and exit. The default is 60 minutes.

Step 7 Click **Next**.

Step 8 Preview the configuration and then click **Create** to create the collection.

Step 9 Configure schedules for the collection job. You can schedule the collection job to run immediately or at specific intervals. For details, see Schedule a collection.

PCEP LSP information is collected and appended to the existing SR-PCE topology, generating an updated network model that includes detailed LSP data.

What to do next

Use this collection as the source network to configure additional collections. For details on configuring different collectors, see the relevant topics in this chapter. For details on editing the collection, see Edit a collection.

Collect multicast flow data from a network

This topic describes how to configure the **Multicast** collector to collect multicast flow data from your network.

The Multicast collector includes these collectors:

- Login find multicast: Logs in to the router to fetch or parse multicast flow data.
- Login poll multicast: Logs in to the router to get multicast traffic rate.
- SNMP find multicast: Collects multicast flow information using SNMP.
- SNMP poll multicast: Collects traffic rate data for multicast flows using SNMP.

Before you begin

Complete the steps mentioned in Preconfiguration workflow.

Procedure

Step 1 Decide whether to create a new collection or edit an existing one. For details, see Configure a collection or Edit a collection.

Step 2 To use an external script as a first step of the collection configuration chain, select the startup script option. When using a startup script, you can either skip the basic topology collector or configure it to use the startup script as its source. If you do not use a startup script, you must select one of the basic topology collectors as needed.

Step 3 In the **Traffic and Demands** section, select **Multicast** and click **Next**.

Step 4 On the Configure page, click **Multicast** in the **Selected collectors** pane on the left.

Note
Ensure that the basic topology parameters are updated to meet your needs. Update the parameters if necessary.

Step 5 Enter these configuration parameters:

- **Source**: Select the source collector whose output serves as the input for this collector.
- **Data collection source**: Select the collector you want to use to collect the multicast data. The options are Login find multicast, Login poll multicast, SNMP find multicast, and SNMP poll multicast.

Step 6 (Optional) Expand the ***Collector* settings** panel and enter the details in the relevant fields. Depending on the collectors you selected in the previous step, the options differ. For descriptions of these advanced options, see Multicast collection advanced options, on page 15.

Step 7 Click **Next**.

Step 8 Preview the configuration and then click **Create** to create the collection.

Step 9 Configure schedules for the collection job. You can schedule the collection job to run immediately or at specific intervals. For details, see Schedule a collection.

The Multicast collector is configured and begins collecting multicast flow data from your network as specified.

What to do next

Use this collection as the source network to configure additional collections. For details on configuring different collectors, see the relevant topics in this chapter. For details on editing the collection, see Edit a collection.

Multicast collection advanced options

You can configure several advanced options when using the Multicast collectors.

Table 4: Multicast collection advanced options

Option	Description
Login find settings	
Data collection timeout	Sets the maximum time allowed for data collection, in minutes. If the specified limit is exceeded, the internal tools used for data collection will time out and exit. The default is 30 minutes.
Use existing config	Uses existing multicast configuration data stored in the cache.
Force config update	Updates multicast configuration files even if they exist in the cache.
Save configs	Saves multicast configurations in the cache or discards them if not selected.

Option	Description
Overwrite files	Overwrites existing configuration files.
Login poll settings	
Data collection timeout	Sets the maximum time allowed for data collection, in minutes. If the specified limit is exceeded, the internal tools used for data collection will time out and exit. The default is 30 minutes.
No of samples	Sets the number of data samples to collect during polling.
Polling interval	Sets the interval, in seconds, between the login rate readings.
Traffic level name	Indicates the name of traffic level.
Traffic filtering	Defines the filtering criteria for multicast traffic from multiple sources for each S|G group.
Use existing config	Uses existing multicast configuration data stored in the cache.
Force config update	Updates multicast configuration files even if they exist in the cache.
Save configs	Saves multicast configurations in the cache or discards them if not selected.
Overwrite files	Overwrites existing configuration files.
SNMP find settings	
Data collection timeout	Sets the maximum time allowed for data collection, in minutes. If the specified limit is exceeded, the internal tools used for data collection will time out and exit. The default is 30 minutes.
SNMP poll settings	
Data collection timeout	Sets the maximum time allowed for data collection, in minutes. If the specified limit is exceeded, the internal tools used for data collection will time out and exit. The default is 30 minutes.
No of samples	Sets the number of data samples to collect during polling.
Polling interval	Sets the interval, in seconds, between the login rate readings.
Traffic level name	Indicates the name of traffic level.
Traffic filtering	Defines the filtering criteria for multicast traffic from multiple sources for each S|G group.
Debug	
Verbosity	Sets the level of detail in log messages. The default is 30, with a valid range from 1 to 60.

Option	Description
Net recorder	Records SNMP messages. The options are Off, Record, and Playback. The default is Off. • Record: The SNMP messages to and from the live network are recorded internally as discovery runs. It is used for debugging. • Playback: The recorded messages are played back through the collector as if they came from the live network, thus providing offline debugging of network collection. • Off: No recording or playback is performed.

Discover BGP peers

This topic describes how to configure the **BGP** collector to discover BGP topology using SNMP and login.

The BGP collector uses a topology network, typically an IGP topology collector output, as its source network and adds BGP links to external ASN nodes.

Before you begin

Complete the steps mentioned in Preconfiguration workflow.

Procedure

Step 1 Decide whether to create a new collection or edit an existing one. For details, see Configure a collection or Edit a collection.

Step 2 To use an external script as a first step of the collection configuration chain, select the startup script option. When using a startup script, you can either skip the basic topology collector or configure it to use the startup script as its source. If you do not use a startup script, you must select one of the basic topology collectors as needed.

Step 3 In the **Advanced modeling** section, select **BGP** and click **Next**.

Step 4 On the Configure page, click **BGP** in the **Selected collectors** pane on the left.

Note
Ensure that the basic topology parameters are updated to meet your needs. Update the parameters if necessary.

Step 5 From the **Source** drop-down list, select the source collector whose output serves as the input for this collector.

Step 6 (Optional) Expand the **Advanced settings** panel and configure any other relevant advanced fields as needed. For descriptions of these advanced options, see BGP topology advanced options, on page 18.

Step 7 Click **Next**.

Step 8 Preview the configuration and then click **Create** to create the collection.

Step 9 Configure schedules for the collection job. You can schedule the collection job to run immediately or at specific intervals. For details, see Schedule a collection.

The BGP collector is now configured and able to discover BGP topology using SNMP and login.

What to do next

Use this collection as the source network to configure additional collections. For details on configuring different collectors, see the relevant topics in this chapter. For details on editing the collection, see Edit a collection.

BGP topology advanced options

You can configure several advanced options when using the BGP collector.

Table 5: BGP topology collection advanced options

Option	Description
ASN include	Specifies ASNs to include. By default, includes all ASNs.
Internal ASNs	Specifies internal ASNs.
Protocol	Specifies the Internet Protocol (IP) versions. The options are IPv4 and IPv6.
Min IPv4 prefix length	Specifies the minimum IPv4 prefix length to control how strictly subnet matching occurs when discovering interfaces as BGP links.
Min IPv6 prefix length	Specifies the minimum IPv6 prefix length to control how strictly subnet matching occurs when discovering interfaces as BGP links.
Login multi hop	Specifies whether to log in to routers that potentially contain multi-hop peers.
Force login platform	Overrides platform detection and uses the specified platform. Valid values are cisco, juniper, alu, huawei.
Fallback login platform	Sets the fallback vendor if platform detection fails. Valid values are cisco, juniper, alu, huawei.
Try send enable	Sends an enable password if the platform type is not detected when logging in to a router. This grants higher-level access required to retrieve or modify configuration on devices that require a secondary “enable password”
Telnet username prompt	Specifies an alternative username prompt for Telnet.
Telnet password prompt	Specifies an alternative password prompt for Telnet.
Find internal ASN links	Finds links between two or more internal ASNs. Normally, this action is not required because IGP discovers these links.
Find non IP exit interface	Searches for exit interfaces that are not represented as next-hop IP addresses, but rather as interfaces (which are rare). **Note** This action increases the amount of SNMP requests for BGP discovery, which affects performance.
Internal exit interface	Discovers BGP links to internal ASNs.

Option	Description
Get MAC address	Collects source MAC addresses of BGP peers connected to an Internet Exchange public peering switch. This action is required only for MAC accounting.
Use DNS	Indicates whether to use DNS to resolve BGP IP addresses.
Force check all	Indicates whether to check all routers even if there is no indication of potential multi-hop peers. This action could be slow.
Data collection timeout	Sets the maximum time allowed for data collection, in minutes. If the specified limit is exceeded, the internal tools used for data collection will time out and exit. The default is 60 minutes.
Debug	
Verbosity	Sets the level of detail in log messages. The default is 30, with a valid range from 1 to 60.
Net recorder	Records SNMP messages. The options are Off, Record, and Playback. The default is Off. • Record: The SNMP messages to and from the live network are recorded internally as discovery runs. It is used for debugging. • Playback: The recorded messages are played back through the collector as if they came from the live network, thus providing offline debugging of network collection. • Off: No recording or playback is performed.
Login record mode	Records the discovery process. The options are Off, Record, and Playback. The default is Off. • Record: The messages to and from the live network are recorded internally as the tool runs. It is used for debugging. • Playback: The recorded messages are played back through the tool as if they came from the live network, thus providing offline debugging of network collection. • Off: No recording or playback is performed.

Discover VPN topology

This topic describes how to configure the **VPN** collector to discover Layer 2 and Layer 3 VPN topology.

Note Currently, only P2P-VPWS xconnect discovery is supported for Layer 2 VPNs.

Before you begin

Complete the steps mentioned in Preconfiguration workflow.

Procedure

Step 1 Decide whether to create a new collection or edit an existing one. For details, see Configure a collection or Edit a collection.

Step 2 To use an external script as a first step of the collection configuration chain, select the startup script option. When using a startup script, you can either skip the basic topology collector or configure it to use the startup script as its source. If you do not use a startup script, you must select one of the basic topology collectors as needed.

Step 3 In the **Advanced modeling** section, select **VPN** and click **Next**.

Step 4 On the Configure page, click **VPN** in the **Selected collectors** pane on the left.

Note
Ensure that the basic topology parameters are updated to meet your needs. Update the parameters if necessary.

Step 5 Enter these configuration parameters:

- **Source**: Select the source collector whose output serves as the input for this collector.
- **VPN type**: Select at least one VPN type.
  - **VPWS**: Select this type when Virtual Private Wire Service (VPWS) is being used in the network.
  - **L3VPN**: Select this type when Layer 3 VPN is being used in the network.

Step 6 (Optional) Expand the **Advanced settings** panel and configure any other relevant advanced fields as needed:

Table 6: VPN collection advanced options

Option	Description
Data collection timeout	Sets the maximum time allowed for data collection, in minutes. If the specified limit is exceeded, the internal tools used for data collection will time out and exit. The default is 60 minutes.
Verbosity	Sets the level of detail in log messages. The default is 30, with a valid range from 1 to 60.
Net recorder	Records SNMP messages. The options are Off, Record, and Playback. The default is Off. • Record: The SNMP messages to and from the live network are recorded internally as discovery runs. It is used for debugging. • Playback: The recorded messages are played back through the collector as if they came from the live network, thus providing offline debugging of network collection. • Off: No recording or playback is performed.

Step 7 Click **Next**.

Step 8 Preview the configuration and then click **Create** to create the collection.

Step 9 Configure schedules for the collection job. You can schedule the collection job to run immediately or at specific intervals. For details, see Schedule a collection.

The VPN collector is now configured.

What to do next

Use this collection as the source network to configure additional collections. For details on configuring different collectors, see the relevant topics in this chapter. For details on editing the collection, see Edit a collection.

Inventory collector and hardware tables

An **Inventory** collector is a Cisco Crosswork Planning component that

- collects hardware inventory information from network devices and
- stores the collected data in structured tables (NetIntHardware*) based on hardware type.

These sections describe the process, components, configuration tables, and best practices for collecting and organizing hardware inventory information using the Inventory collector.

NetIntHardware tables

NetIntHardware* tables store the collected hardware information based on hardware type.

These are a few examples of NetIntHardware tables:

- NetIntHardwareChassis: stores router chassis objects identified by node IP address and SNMP ID.
- NetIntHardwareContainer: stores slot entries 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: stores information about hardware devices that can be installed into other hardware devices. Generally, these devices directly support traffic such as line cards, modules, and route processors, and do not fall into one of the other function-specific hardware tables.
- NetIntHardwarePort: stores physical ports on routers.

Hardware hierarchy

A hardware has a parent-child relationship based on where the object resides within the router. The chassis, considered as the *root object*, has no parent. Except chassis, every object has one parent and can have multiple child objects. Objects without children, such as ports and empty containers, are called *leaf objects*. Hardware hierarchy generally reflects how hardware objects are installed within other objects. For instance, a module representing a line card can have as its parent a container that represents 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 of 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	NetIntHardware Chassis	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. Using this trace, the Inventory collector determines how to process the hardware devices. You must use this process when adding an entry to the HWInventoryTemplates table.

NetIntNodeInventory table

The Inventory collector constructs the NetIntNodeInventory table by processing the NetIntHardware* tables. The collector requires two configuration files and can additionally use an optional one.

- Template file (required): This file contains these tables.
  - HWInventoryTemplates: Contains entries that categorize the devices in the final NetIntNodeInventory table, as well as prune from inclusion.
  - HWNameFormatRules: Contains entries that format the hardware object names to make them more usable, as well as correct unexpected SNMP results.
- Exclude file (required): Contains the ExcludeHWList table that prevents (blocked lists) 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.
- Hardware spec file (optional): Contains the HardwareSpec table that can be used to adjust collected data in terms of the number of slots in a specified device when the slots returned by SNMP is inaccurate.

If you modify the template or choose to exclude files, ensure these changes persist across software upgrades.

HWInventoryTemplates and HWNameFormatRules tables

The **Template file** option under the **Build inventory options** section calls a file containing both the HWInventoryTemplates and the HWNameFormatRules tables.

HWInventoryTemplates Table

The HWInventoryTemplates table tells the Inventory collector how to interpret hardware referenced by the NetIntHardware* tables. It enables the Inventory collector 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, line card, 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 line card. All other hardware objects are categorized by their same name. For instance, a chassis is categorized as a chassis.

The Inventory collector looks at these 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** option. In this instance, if no matches are found using the first two criteria, Cisco Crosswork Planning 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 the Inventory collector how to categorize the hardware. These latter columns identify hardware object types: chassis, slot, line card, module slot, module, port slot, port, or transceiver. Each column entry has the *Type,Identifier,Name* format.

1. Type is the discovered hardware type, such as “container.”
2. Identifier specifies which object (of one or more of the same type) in the hierarchy is referenced (0, 1, ...).
3. Name specifies a column heading in the NetIntHardware* table. This is the name that appears in for that object in the NetIntNodeInventory table . For example, Module,0,Model. "Model" is a column heading in the NetIntHardwareModule table.

You can specify multiple name source columns using a colon. For example, Container,0,Model:Name

If a hardware category does not exist or is empty, the Inventory collector excludes it from the final NetIntNodeInventory table.

Example:

Using the first row of the default Template file, the Cisco Crosswork Planning collector searches the NetIntHardware* tables for ones that have entries that match the Vendor, Model, and DiscoveredHWHierarchy columns as 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 line card, 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.

Add HWInventoryTemplates Entries

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

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

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

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

The entries in the table work as follows:

1. 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.
2. Replaces all Cisco linecard names that match 800-20017-.* with 1X10GE-LR-SC.
3. Replaces all Juniper chassis named "Juniper (MX960) Internet Backbone Router" with MX960.

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.

Exclude Hardware by Model or Name

The **Exclude file** option under the **Build inventory options** section option calls a file containing the ExcludeHWList table . 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:

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

Table entries function as described::

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

HardwareSpec

The **Hardware spec file** option under the **Build inventory options** section calls a file containing the HardwareSpec table . 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, line card, or module. SlotNum indicates the slot number range. For instance, some routers start with slot 0, whereas others start with slot 1.

Example:

HardwareSpec				
Vendor	**HWType**	**Model**	**TotSlot**	**SlotNum**
Cisco	Chassis	7609	9	1–9

Configure inventory collection

This topic describes how to configure the **Inventory** collector.

Before you begin

Complete the steps mentioned in Preconfiguration workflow.

Procedure

Step 1 Decide whether to create a new collection or edit an existing one. For details, see Configure a collection or Edit a collection.

Step 2 To use an external script as a first step of the collection configuration chain, select the startup script option. When using a startup script, you can either skip the basic topology collector or configure it to use the startup script as its source. If you do not use a startup script, you must select one of the basic topology collectors as needed.

Step 3 In the **Traffic and Demands** section, select **Inventory** and click **Next**.

Step 4 On the Configure page, click **Inventory** in the **Selected collectors** pane on the left.

Note
Ensure that the basic topology parameters are updated to meet your needs. Update the parameters if necessary.

Step 5 From the **Source** drop-down list, select the source collector whose output serves as the input for this collector.

Step 6 (Optional) Expand the **Advanced settings** panel and configure any other relevant advanced fields as needed. For descriptions of these advanced options, see Inventory collection advanced options, on page 26.

Step 7 Click **Next**.

Step 8 Preview the configuration and then click **Create** to create the collection.

Step 9 Configure schedules for the collection job. You can schedule the collection job to run immediately or at specific intervals. For details, see Schedule a collection.

The Inventory collector is now configured according to your settings.

What to do next

Use this collection as the source network to configure additional collections. For details on configuring different collectors, see the relevant topics in this chapter. For details on editing the collection, see Edit a collection.

Inventory collection advanced options

You can configure several advanced options when using the Inventory collector.

Table 7: Inventory collection advanced options

Option	Description
Get inventory options	
Login allowed	Allows logging in to the router to collect inventory data.
Data collection timeout	Sets the maximum time allowed for data collection, in minutes. If the specified limit is exceeded, the internal tools used for data collection will time out and exit. The default is 30 minutes.
Build inventory options	

Option	Description
Exclude file	Allows you to select the file that contains the ExcludeHWList table. This table defines hardware characteristics to match against for exclusion in the output. Click the **Download sample file** link to download a sample file that contains the ExcludeHWList table.
Guess template	Broadens the search when processing raw inventory data.
Template file	Allows you to select the hardware template file that contains the HWInventory Templates and HWNameFormatRules tables. Click the **Download sample file** link to download a sample template file.
Hardware spec file	Allows you to select the file that contains the HardwareSpec table. This table defines slot counts for specific types of hardware to verify SNMP data returned from routers. Click the **Download sample file** link to download a sample file that contains the HardwareSpec table.
Debug	
Verbosity	Sets the level of detail in log messages. The default is 30, with a valid range from 1 to 60.
Net recorder	Records SNMP messages. The options are: Off, Record, and Playback. The default is Off. • Record: The SNMP messages to and from the live network are recorded internally as discovery runs. It is used for debugging. • Playback: The recorded messages are played back through the collector as if they came from the live network, thus providing offline debugging of network collection. • Off: No recording or playback is performed.

Collect port, LSP, SRLG, and VPN information using configuration parsing

This topic describes how to configure the **Config parsing** collector to collect port, LSP, SRLG, and VPN information.

Note The **Config parsing** collector is not a base topology collector. Use it only to augment details that are missing from other methods of collection, such as SNMP and SR-PCE.

Before you begin

Complete the steps mentioned in Preconfiguration workflow.

Procedure

Step 1 Decide whether to create a new collection or edit an existing one. For details, see Configure a collection or Edit a collection.

Step 2 To use an external script as a first step of the collection configuration chain, select the startup script option. When using a startup script, you can either skip the basic topology collector or configure it to use the startup script as its source. If you do not use a startup script, you must select one of the basic topology collectors as needed.

Step 3 In the **Advanced modeling** section, select **Config parsing** and click **Next**.

Step 4 On the Configure page, click **Config parsing** in the **Selected collectors** pane on the left.

Note
Ensure that the basic topology parameters are updated to meet your needs. Update the parameters if necessary.

Step 5 From the **Source** drop-down list, select the source collector whose output serves as the input for this collector.

Step 6 Expand the **Get config** and **Parse config** panels. Enter the details in the relevant fields. For field descriptions, see Configuration parsing advanced options, on page 28.

Note
- L2VPN config parse is not supported.
- When L3VPN information is collected by the Config Parsing collector, all VPNs are assumed to be connected to each other.
- If both the Config Parsing collector and the VPN collector are collecting VPN information, ensure that the VPN collector runs before the Config Parsing collector in the collector chain.
- Single-ended SRLGs with a missing end are collected via SR-PCE. The SRLGSCircuits table is not updated for these entries.

Step 7 Click **Next**.

Step 8 Preview the configuration and then click **Create** to create the collection.

Step 9 Configure schedules for the collection job. You can schedule the collection job to run immediately or at specific intervals. For details, see Schedule a collection.

What to do next

Use this collection as the source network to configure additional collections. For details on configuring different collectors, see the relevant topics in this chapter. For details on editing the collection, see Edit a collection.

Configuration parsing advanced options

You can configure several advanced options when using the Config parsing collector.

Table 8: Configuration parsing advanced options

Option	Description
Get config options	
Collect configuration	Retrieves configuration details from devices or routers.
Force login platform	Overrides platform detection and uses the specified platform. Valid values are cisco, juniper, alu, huawei.
Fallback login platform	Sets the fallback vendor in case platform detection fails. Valid values are cisco, juniper, alu, huawei.
Try send enable	Sends an enable password if the platform type is not detected when logging in to a router. This grants higher-level access required to retrieve or modify configuration on devices that require a secondary “enable password”.
Telnet username prompt	Specifies an alternative username prompt for Telnet.
Telnet password prompt	Specifies an alternative password prompt for Telnet.
Data collection timeout	Sets the maximum time allowed for data collection, in minutes. If the specified limit is exceeded, the internal tools used for data collection will time out and exit. The default is 60 minutes.
Parse config options	
Protocol type	Allows you to select the IGP protocol running in the network. The options are isis, ospf, and None. The default is **isis**.
ISIS level	Indicates the ISIS level to use. The agent can read IS-IS Level 1, Level 2, or both. If you select both, the agent combines both levels into a single network and Level 2 metrics take precedence.
OSPF area	Specifies whether to collect a single OSPF area or all areas. This option specifies the area ID or all. The default is area 0.
ASN	Specifies the ASN to collect. ASN is ignored by default. However, for networks that span multiple BGP ASNs, use this option to read information from more than one IGP process ID or instance ID in an ASN.
Include objects	Allows you to select the configuration objects that you want to parse. The available options are LAG, SRLG, RSVP and CS RSVP, VPN, FRR, SR LSPS, LMP, and SR Policies.
Circuit match	Indicates the criteria to use to form circuits.
LAG port match	Controls how to match local and remote ports in port circuits. • Guess: Creates port circuits to match as many ports as possible. • None: Does not create port circuits.

Option	Description
OSPF process ID	Specifies which OSPF process ID to use when there are multiple OSPF processes.
IS-IS instance ID	Specifies which IS-IS instance ID to use when there are multiple IS-IS instances.
Loopback interface	Specifies the loopback interface number to use for the router IP.
Resolve references	Enables resolution of IP address references during parsing.
Multithreading	Enables multithreaded processing of configuration files to speed up parsing.
Filter showcommands	Filters multiple show commands.
Build topology	Constructs network topology after parsing the configuration.
Shared media	Creates pseudonodes for shared media.
Data collection timeout	Sets the maximum time allowed for data collection, in minutes. If the specified limit is exceeded, the internal tools used for data collection will time out and exit. The default is 60 minutes.
Debug	
Verbosity	Sets the level of detail in log messages. The default is 30, with a valid range from 1 to 60.
Net recorder	Records SNMP messages. The options are Off, Record, and Playback. The default is Off. • Record: The SNMP messages to and from the live network are recorded internally as discovery runs. It is used for debugging. • Playback: The recorded messages are played back through the collector as if they came from the live network, thus providing offline debugging of network collection. • Off: No recording or playback is performed.

Collect Circuit Style RSVP-TE information

This topic describes how to collect Circuit Style RSVP (CS-RSVP) LSP information from network devices.

Circuit Style RSVP (CS-RSVP) LSPs are logical entities that bundle two unidirectional RSVP LSPs with the same endpoints to form bidirectional RSVP LSPs. This allows traffic to consistently travel in both directions between the endpoints.

To collect CS RSVP-TE data, you must configure the **LSP** and **Config parsing** collectors. The Config parsing collector is required to collect the configuration data from each device in the network and parse the CS-RSVP data out of it. After the collection is run successfully, the aggregated plan file includes the CS-RSVP LSP details collected from the devices.

Before you begin

- Complete the steps mentioned in Preconfiguration workflow.
- Ensure that the devices have these configurations:
  - RSVP configuration with **bidirectional** enabled.
  - The **bidirectional** configuration includes the same **association id**, **source-address**, and **global-id** in both directions.
  - The **bidirectional** configuration specifies the **association type** as **co-routed**.

Procedure

Step 1 Decide whether to create a new collection or edit an existing one. For details, see Configure a collection or Edit a collection.

Step 2 To use an external script as a first step of the collection configuration chain, select the startup script option. When using a startup script, you can either skip the basic topology collector or configure it to use the startup script as its source. If you do not use a startup script, you must select one of the basic topology collectors as needed.

Step 3 In the **Advanced modeling** section, select the **LSP** and **Config parsing** collectors. Then, click **Next**.

Step 4 Configure both the **LSP** and **Config parsing** collectors. Ensure to select the **RSVP and CS RSVP** option from the **Include objects** drop-down list. This option is available in the **Parse config** section of the **Config parsing** page.

For details on the other LSP and Config parsing options, see Collect LSP information, on page 11 and Collect port, LSP, SRLG, and VPN information using configuration parsing, on page 27.

Note
To retrieve the signaled name, ensure the LSP collector executes before the Config parsing collector. If this order is not followed, LSP tunnel name collected by Config parsing will overwrite the signaled name value collected by the LSP collector.

Step 5 (Optional) Expand the **Advanced settings** panel and configure any other relevant fields. For descriptions of these advanced options, see LSP collection advanced options, on page 12.

Step 6 Click **Next**.

Step 7 Preview the configuration and then click **Create** to create the collection.

Step 8 Configure schedules for the collection job. You can schedule the collection job to run immediately or at specific intervals. For details, see Schedule a collection.

The resulting network model includes the CS-RSVP LSP details.

What to do next

Use this collection as the source network to configure additional collections. For details on configuring different collectors, see the relevant topics in this chapter. For details on editing the collection, see Edit a collection.

Configure the Layout collector for improved network model visualization

This topic describes how to configure the **Layout** collector.

The **Layout** collector adds layout properties to a source network model. This improves visualization when you import the plan file into Cisco Crosswork Planning. The collector automatically records changes to the layout properties. When the source network model changes, the layout of the destination model is updated.

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

Note The Layout collector saves only the node and site mappings. It does not save the node's coordinates.

Before you begin

Complete the steps mentioned in Preconfiguration workflow.

Procedure

Step 1 Decide whether to create a new collection or edit an existing one. For details, see Configure a collection or Edit a collection.

Step 2 To use an external script as a first step of the collection configuration chain, select the startup script option. When using a startup script, you can either skip the basic topology collector or configure it to use the startup script as its source. If you do not use a startup script, you must select one of the basic topology collectors as needed.

Step 3 In the **Traffic and Demands** section, select **Layout** and click **Next**.

Step 4 On the Configure page, click **Layout** in the **Selected collectors** pane on the left.

Note
Ensure that the basic topology parameters are updated to meet your needs. Update the parameters if necessary.

Step 5 Enter these configuration parameters:

- **Source**: Select the source collector whose output serves as the input for this collector.
- **Template file**: Enter the template plan file path from where the layout details are copied.

  Note
  If you are migrating the collector configuration either from Cisco WAE or from another Cisco Crosswork Planning instance, ensure that the **Template file** field is updated with the correct file after importing the collector configuration. This is necessary because, after importing the configuration, the server restores only the file name and not the actual file. If the field is not updated with the correct file, then the collection fails.

Step 6 (Optional) Expand the **Advanced settings** panel and enter the following information:

- **Timeout**: Enter the maximum time allowed for data collection, in minutes. If the specified limit is exceeded, the internal tools used for data collection will time out and exit. The default is 60 minutes.

Step 7 Click **Next**.

Step 8 Preview the configuration and then click **Create** to create the collection.

Step 9 Configure schedules for the collection job. You can schedule the collection job to run immediately or at specific intervals. For details, see Schedule a collection.

The Layout collector is now configured.

What to do next

Use this collection as the source network to configure additional collections. For details on configuring different collectors, see the relevant topics in this chapter. For details on editing the collection, see Edit a collection.

Collect traffic statistics

This topic describes how to configure the **Traffic collection** collector.

The **Traffic collection** collector collects traffic statistics, such as interface traffic, LSP traffic, MAC traffic, and VPN traffic using SNMP polling. After configuring the **Traffic collection** collector, you can view the traffic poller agent details in the **Collector** > **Agents** page. The agent name matches the collection name.

Note During the first traffic collection run, the traffic data is not populated in the plan file due to insufficient data to compute traffic details. Beginning with the second or third run, depending on the schedule duration and the configuration of minimum and maximum window lengths, traffic data begins to populate in the plan file.

Before you begin

- Complete the steps mentioned in Preconfiguration workflow.
- To collect VPN traffic, you must have a VPN network model. For details, see Discover VPN topology, on page 19.
- To collect LSP traffic, you must have an LSP network model. For details, see Collect LSP information, on page 11.

Procedure

Step 1 Decide whether to create a new collection or edit an existing one. For details, see Configure a collection or Edit a collection.

Step 2 To use an external script as a first step of the collection configuration chain, select the startup script option. When using a startup script, you can either skip the basic topology collector or configure it to use the startup script as its source. If you do not use a startup script, you must select one of the basic topology collectors as needed.

Step 3 In the **Traffic and Demands** section, select **Traffic collection** and click **Next**.

Step 4 On the Configure page, click **Traffic collection** in the **Selected collectors** pane on the left.

Note
Ensure that the basic topology parameters are updated to meet your needs. Update the parameters if necessary.

a) Check the **Traffic collection** check box to enable the traffic poller.
b) From the **Source** drop-down list, select the source collector whose output serves as the input for this collector.
c) To run continuous traffic collection for interfaces, enable **Interface traffic poll** and then enter the following:
   - **Polling period**: Enter the polling period in seconds. We recommend starting with 60 seconds.
   - **QoS**: Check the **Enable** check box if you want to enable queues traffic collection.
   - **VPN**: Check the **Enable** check box if you want to enable VPN traffic collection. If enabled, confirm that the source network model has VPNs enabled.
d) To run continuous traffic collection for LSPs, enable **LSP traffic poll** and then enter the following:
   - **Polling period**: Enter the polling period in seconds. We recommend starting with 60 seconds.

   Note
   If **LSP traffic poll** is enabled, make sure that the source network model has all the LSP details.
e) To run continuous traffic collection for MAC accounting, enable **MAC traffic poll** and then enter the following:
   - **Polling period**: Enter the polling period in seconds. We recommend starting with 60 seconds.

   Note
   If **MAC traffic poll** is enabled, make sure that the source network model has MAC addresses.
f) (Optional) Expand the **SNMP traffic computation** panel and enter the details in the relevant fields. For field descriptions, see Traffic collection advanced options, on page 34.

Step 5 Click **Next**.

Step 6 Preview the configuration and then click **Create** to create the collection.

Step 7 Configure schedules for the collection job. You can schedule the collection job to run immediately or at specific intervals. For details, see Schedule a collection.

The traffic details are updated in the plan files only on running the scheduled jobs. If a job is not executed, the traffic data is not updated in the plan files.

Traffic statistics are collected and available in the resulting plan file on execution of the subsequent scheduled jobs.

What to do next

Use this collection as the source network to configure additional collections. For details on configuring different collectors, see the relevant topics in this chapter. For details on editing the collection, see Edit a collection.

Traffic collection advanced options

You can configure several advanced options when using Traffic collection.

Table 9: Traffic collection advanced options

Option	Description
Minimum window length	Specifies the minimum window length for traffic calculation, in seconds. The default is 300 seconds.
Maximum window length	Specifies the maximum window length for traffic calculation, in seconds. The default is 450 seconds.
Raw counter TTL	Determines how long raw counter data is kept, measured in minutes. The default is 15 minutes.
Discard over capacity	Discards traffic rates that are higher than capacity.
Net recorder file max size	Specifies the maximum size for the net record file.
Data collection timeout	Sets the maximum time allowed for data collection, in minutes. If the specified limit is exceeded, the internal tools used for data collection will time out and exit. The default is 60 minutes.
Debug	
Verbosity	Sets the level of detail in log messages. The default is 30, with a valid range from 1 to 60.
Net recorder	Records SNMP messages. The options are: Off, Record, and Playback. The default is Off. • Record: The SNMP messages to and from the live network are recorded internally as discovery runs. It is used for debugging. • Playback: The recorded messages are played back through the collector as if they came from the live network, thus providing offline debugging of network collection. • Off: No recording or playback is performed.

Tune the traffic poller settings

This topic describes the efficient way to run traffic polling.

The traffic poller collects raw traffic counters from the network. The collection time depends on network size, network latency, and response time from individual nodes.

Procedure

Step 1 Set the traffic poller verbosity to 40 in the **Traffic collection** configuration page.

Step 2 Start with the default options and run continuous collection for several hours. The default values include:

```
Interface traffic poll > Polling period = 60
LSP traffic poll > Polling period = 60
```

```
Minimum window length = 300
Maximum window length = 450
Raw counter TTL = 15
```

Step 3 Configure the Traffic collection scheduler to run every 300 seconds.

Step 4 Download the `continuous_poller_out.log` file using the showtech option.

a) From the main menu, choose **Administration** > **Crosswork Manager** > **Crosswork Health** > **Collector**.
b) Click the **Microservices** tab.
c) Click ••• for the **collection-service** and choose **Request logs**.
d) Download the resulting tar file to view the log file.

Step 5 Search for actual collection times.

Example:

```
Info [40]: LSP Traffic Poller: Collection complete. Duration: 43.3 sec
Info [40]: Interface Traffic Poller: Collection complete. Duration: 42.7 sec
```

In this example, the fastest pace at which the poller can poll the network is around 40 to 50 seconds. This value represents the minimum polling period for both Interface traffic poll and LSP traffic poll. Since the traffic poller populates traffic for both interfaces and LSPs at the same time, set both values to the same number.

The traffic poller calculates traffic by collecting raw traffic counters, such as c1, c2, and so on. It requires at least two counters to calculate traffic.

```
(c2.counter - c1.counter)/(c2.timestamp - c1.timestamp)
```

Best practices for poller configuration

Follow these best practices to optimize poller configuration and ensure reliable traffic data collection:

- Set **Minimum window length** to at least **2 * polling period**, because the poller requires at least two counters. Increase the window length by 25% or more to accommodate network variations.

  Minimum window length is a sliding window that is used to sample two counters. It looks for two counters which are farthest apart, that is, the latest and earliest within a specified period. The average traffic is calculated for this period. Since the poller requires at least two counters, the smallest value must be at least twice the polling period.

- Set **Maximum window length** to at least **2 * polling period**. Increase the window length by 50% or more to accommodate network variations. For unresponsive nodes, increase by 100% or more.

  If the Minimum window length does not find enough counters for the specified period due to increased network latency or node response time, the poller reports traffic as N/A. To avoid empty traffic data, use an insurance window called **Maximum window length**.

- Set **Raw counter TTL** to be equal to or greater than **Maximum window length**.

  The traffic poller stores raw counters in memory for traffic calculation, which uses RAM space. The traffic poller periodically cleans up old counters stored in memory. The system deletes any counter data older than Raw counter TTL (minutes).

- Monitor the poller memory usage and the time taken to populate traffic. Traffic population in traffic poller is the process of calculating traffic in the network and updating the plan file. The duration depends on network size. The system logs the actual time taken to populate traffic in the snmp-traffic-poller-service.log file.

Lines from an example log file:

```
TrafficCalculatorRfs Did-52-Worker-46: - Traffic calculation took (ms) 379976
TrafficCalculatorRfs Did-52-Worker-46: - Traffic calculation took (ms) 391953
TrafficCalculatorRfs Did-52-Worker-46: - Traffic calculation took (ms) 388853
```

In this example, the fastest rate for populating traffic, and consumed by other tools, is about 400 seconds.

- If you see "Invalid counter" warnings in the snmp-traffic-poller-service.log file, for example, c1.counter is greater than c2.counter resulting in negative traffic, recognize that counters might have reset or overflowed. This issue commonly occurs with 32-bit counters. If you see many such errors, increase the sliding window sizes to process more counters and reduce chance of failure.
- Do not poll network at a faster rate than populating traffic. In the example above, the most aggressive polling setting is 50 seconds, whereas traffic population takes around 400 seconds. This results in eight wasted network polls. To resolve this, increase the traffic polling period, sliding window sizes, and Raw counter TTL.
  - Here is the recommended configuration for this example:
    1. Set these values:

       ```
       Interface traffic poll > Polling period 180
       LSP traffic poll enabled
       LSP traffic poll > Polling period 180
       Minimum window length 400
       Maximum window length 800
       Raw counter TTL 15
       Data collection timeout 60
       ```

    2. Configure Traffic collection scheduler to run every 400 seconds.

Note Data collection timeout is adjusted to 60 minutes for traffic population. This timeout is not used generally and should be set only as high as necessary.

  - You can adjust these numbers to be less aggressive to save CPU resources and network bandwidth. To do this:
    1. Set these values:

       ```
       Interface traffic poll > Polling period 240
       LSP traffic poll enabled
       LSP traffic poll > Polling period 240
       Minimum window length 600
       Maximum window length 1200
       Raw counter TTL 20
       Data collection timeout 60
       ```

    2. Configure Traffic collection scheduler to run every 600 seconds.

Collect traffic demands information

This topic describes how to configure the **Demand deduction** collector to collect information about traffic demands from the network.

Before you begin

Complete the steps mentioned in Preconfiguration workflow.

Procedure

Step 1 Decide whether to create a new collection or edit an existing one. For details, see Configure a collection or Edit a collection.

Step 2 To use an external script as a first step of the collection configuration chain, select the startup script option. When using a startup script, you can either skip the basic topology collector or configure it to use the startup script as its source. If you do not use a startup script, you must select one of the basic topology collectors as needed.

Step 3 In the **Traffic and Demands** section, select **Demand deduction** and click **Next**.

Step 4 On the Configure page, click **Demand deduction** in the **Selected collectors** pane on the left.

Note
Ensure that the basic topology parameters are updated to meet your needs. Update the parameters if necessary.

Step 5 From the **Source** drop-down list, select the source collector whose output model serves as the input for this collector.

Step 6 Under **Demand mesh steps**, click **+ Add step** to add a step.

On the Add Mesh Step page, enter these details:

a) In the **Name** field, enter the name for the step.
b) In the **Step number** field, enter the execution order for this step.
c) From the **Tool** drop-down list, select the required tool. The available tools include Demands for P2MP LSPs, Demand deduction, External executable script, Copy demands, Demands for LSPs, or Demand mesh creator.
d) Check the **Enable** check box to run the selected tools.
e) Update or enter the details in the **Tool configuration** section. The options differ based on the selected tool.
f) (Optional) Expand the **Advanced** panel and enter the relevant details.
g) Click **Continue**.

Repeat this step to add more steps to the configuration.

To remove any of the steps added, select the step and click the **Delete** button at the bottom of the Add Mesh Step page.

Step 7 Click **Next**.

Step 8 Preview the configuration and then click **Create** to create the collection.

Step 9 Configure schedules for the collection job. You can schedule the collection job to run immediately or at specific intervals. For details, see Schedule a collection.

The Demand deduction collector is configured to collect information about traffic demands from the network.

What to do next

Use this collection as the source network to configure additional collections. For details on configuring different collectors, see the relevant topics in this chapter. For details on editing the collection, see Edit a collection.

NetFlow data collection

NetFlow data collection is a process where Cisco Crosswork Planning

- collects NetFlow and related flow measurements from the network devices
- aggregates these measurements to construct accurate demand traffic data for Cisco Crosswork Planning Design, and
- provides an alternative to estimating the demand traffic from interfaces, LSPs, and other statistics using Demand deduction.

A NetFlow collector gathers information about traffic flow and helps build a traffic and demand matrix.

Importing flow measurements is useful when there is full or nearly complete flow coverage at a network's edge routers. In addition, it is beneficial when accuracy of individual demands between external autonomous systems (ASes) is of interest.

Data collected separately by the collectors, such as topology, BGP neighbors, and interface statistics, combines with flow measurements to scale flows and provide a complete demand mesh between both external ASes and internal nodes.

Note If the NetFlow collector is part of multiple collections, you cannot execute those collections at the same time. You must run each collection individually because the NetFlow collector does not support simultaneous execution of collections.

Types of data collected

Cisco Crosswork Planning gathers these types of data to build a network model with flows and their traffic measurements aggregated over time:

- Flow traffic using NetFlow, JFlow, CFlowd, IPFIX, and NetStream flows
- Interface traffic and BGP peer information collected over SNMP
- BGP path attributes over peering sessions

NetFlow collection configuration requirements

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.

The configuration requirements include:

- Configure routers to export flows and establish BGP peering with the flow collection server.
- Export NetFlow v5, v9, and IPFIX datagrams to the UDP port of the flow collection server, which has a default setting of 2100. Export of IPv6 flows requires NetFlow v9 or IPFIX.
- Define a BGP session on the routers configured as iBGP Route Reflector Client for the flow collector server. If configuring this in the router itself is not feasible, then a BGP Route Reflector Server with a complete view of all relevant routing tables can be used instead.

- Configure the source IPv4 address of flow export datagrams 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.
- Limit the maximum length of the BGP AS path attribute to three hops when receiving BGP routes. This helps to prevent excessive server memory consumption. The total length of BGP attributes attached to a single IP prefix, including AS path, can be very large, up to 64 KB.

Configure the NetFlow collection

This topic describes how to configure the **NetFlow** collector.

Before you begin

- Complete the steps mentioned in Preconfiguration workflow.
- Ensure all NetFlow agents are configured to operate in single mode.

Procedure

Step 1 Decide whether to create a new collection or edit an existing one. For details, see Configure a collection or Edit a collection.

Step 2 To use an external script as a first step of the collection configuration chain, select the startup script option. When using a startup script, you can either skip the basic topology collector or configure it to use the startup script as its source. If you do not use a startup script, you must select one of the basic topology collectors as needed.

Step 3 In the **Traffic and Demands** section, select **NetFlow**, and click **Next**.

Step 4 On the Configure page, click **NetFlow** in the **Selected collectors** pane on the left.

Note
Ensure that the basic topology parameters are updated to meet your needs. Update the parameters if necessary.

Step 5 Enter these configuration parameters:

- **Source**: Select the source collector whose output serves as the input for this collector.
- **Agents**: Select the applicable agents from the drop-down list.

Step 6 In the **Common config** section, from the **Split AS flows on ingress** drop-down list, select the traffic aggregation strategy for external ASNs.

(Optional) Enter information in the other fields. For field descriptions, see NetFlow collection advanced options, on page 41.

Step 7 (Optional) Expand the **IAS flows** and **Demands** panels, and configure any other relevant advanced fields as needed. For descriptions of these options, see NetFlow collection advanced options, on page 41. Then, click **Next**.

Step 8 Preview the configuration and then click **Create** to create the collection.

Step 9 Configure schedules for the collection job. You can schedule the collection job to run immediately or at specific intervals. For details, see Schedule a collection.

The NetFlow collection configuration is now complete.

What to do next

Use this collection as the source network to configure additional collections. For details on configuring different collectors, see the relevant topics in this chapter. For details on editing the collection, see Edit a collection.

NetFlow collection advanced options

You can configure several advanced options when using the NetFlow collector.

Table 10: NetFlow collection advanced options

Option	Description
Common config	
Split AS flows on ingress	Specifies the traffic aggregation strategy for external ASNs. When multiple external ASNs are connected to an IXP switch, it determines whether to aggregate traffic data from all ASNs or to distribute it proportionally to MAC accounting ingress traffic.
ASN	Specifies the ASN of the internal AS in the network.
Address family	Specifies the list of protocol versions to include. Enter the versions as a comma-separated list.
Ext node tags	Allows you to enter one or more node tags. Click + to add multiple node tags.
Split AS flows on egress	Splits Inter AS flows as they exit the network through all the interfaces connected to the egress AS.
Extra aggregation	Allows you to select additional aggregation keys from the drop-down list.
Log level	Specifies the log level of the tool. The options are Off, Fatal, Error, Warn, Notice, Info, Debug, and Trace.
Number of threads	Specifies the maximum number of threads to be used in parallel computation.
IAS flows	
Trim inter AS flows	Specifies the value in MBits/sec below which the Inter AS flows for traffic is strictly discarded.
Match BGP external info	Specifies whether to match egress IP addresses in the BGP peer relation.
Ingress interface filter	Specifies a filter of node and interface in the form Node:InterfaceName that will be applied while reading the flow matrix to filter only those ingress interfaces.
Egress interface filter	Specifies a filter of node and interface in the form Node:InterfaceName that will be applied while reading the flow matrix to filter only those egress interfaces.

Option	Description
Back track micro flows	Specifies whether to generate files showing a relationship between micro flows from the input file and those demands or Inter AS flows that aggregate them.
Flow import IDs	Allows you enter comma separated flow IDs to import data from.
IAS computation timeout	Specifies the timeout for IAS flows computation, in minutes. The valid range is 1 to 1440. The default is 60 minutes.
Demands	
Demand name	Specifies the name for any new demands.
Demand tag	Specifies the tag for any new demands, or to append to the existing demands.
Trim demands	Discards demands below a set threshold (in Mbits/sec).
Demand service class	Specifies the service class for demands.
Demand traffic level	Specifies the traffic level for demands.
Missing flows	Specifies the path where the file with interfaces that are missing flows is generated.

Run an external script against a network model

This topic describes how to run an external script against a network model.

The external scripts let you run a customized script against a selected network model. Use this feature when you need specific data from your network that the existing collectors cannot provide. In this case, you take an existing collection model created in Cisco Crosswork Planning and append information from a custom script to create a final network model that contains the data you need.

For an example of a custom script, see Sample script for updating interface descriptions, on page 45.

Before you begin

- Complete the steps mentioned in Preconfiguration workflow.
- Have the custom script and any supporting files ready in one of the accepted file formats or compressed archives.

Note If you are using a script from Cisco WAE and intend to use it within Cisco Crosswork Planning, it may not function as expected without modifications. This is due to architectural differences between Cisco WAE and Cisco Crosswork Planning, including variations in how the files are referenced. You must adjust the script appropriately for use in Cisco Crosswork Planning.

Procedure

Step 1 Decide whether to create a new collection or edit an existing one. For details, see Configure a collection or Edit a collection.

Step 2 Select one of the basic topology collectors, as needed. Optionally, select other advanced collectors according to your needs.

Step 3 On the Configure page, click **+ Add external script** under the Basic topology, Advanced modeling, or Traffic and Demands section.

Step 4 Enter these details:

- **Collector name**: Specify the name for this collection.
- **Is source a plan file?**: Check this check box if you want to run the script on a plan file. If you select this option, enter the plan file details in the **Input plan file** field.
- **Source**: Select the collector on which you want to run the external script. For example, if you select BGP as the Source, the custom script is executed on the BGP collector. The output model from the BGP collection is updated based on the specifications mentioned in the custom script. You also have the option to select DARE or SAgE aggregator output as the source. Any script having source as SAgE is executed after the SAgE aggregation and archival tasks.
- **Input file**: Upload your custom script along with any supporting files necessary for its successful execution. If multiple files are required, compress them into a single archive before uploading. Valid file formats are .py, .sh, .pl, .zip, .tar, .gz, and .tar.gz.

  Note
  Each time a file is uploaded, the input file option is overwritten.

- **Executable script**: Enter the name of the file that initiates the script execution process. This is one of the files you uploaded in the **Input file** field.

  The external script executor provides command line arguments that enable custom scripts to access specific files and the home directory. The arguments are predefined and follow a specific order. Understanding what each argument represents is important to ensure proper usage.

  These are the details of the arguments:

  - argv[1]: Source plan file
  - argv[2]: Output plan file
  - argv[3]: Device access authentication file
  - argv[4]: Global network access configuration file
  - argv[5]: Home directory
  - argv[6]: Path where user uploaded external files are available
  - argv[7]: Path to access archive root directory

  Example:

  The Sample script for updating interface descriptions, on page 45 appends a description to every interface in the network with "My IGP metric is *value*" based on the data from an Excel file named "description.xlsx". The argv[5]

parameter in the script specifies the home directory path of the "description.xlsx" file. For the script to run successfully, note that you must include this Excel file in the compressed file before uploading via the **Input file** field.

- **Script language**: Select the language of the custom script. The valid script languages are Python, Shell, and Perl.
- **Aggregator properties**: If you want to specify any tables or columns to be aggregated, then list them in a .properties file and upload the file using this field. By default, all columns and tables are aggregated.
- **Timeout**: Specify the action timeout. The default is 30 minutes.

Step 5 Click **Next**.

Step 6 Preview the configuration and then click **Create** to create the collection.

Step 7 Configure schedules for the collection job. You can schedule the collection job to run immediately or at specific intervals. For details, see Schedule a collection.

The custom script executes against the selected network model.

Accessing dynamic data files through external scripts

Summary

Cisco Crosswork Planning allows you to upload data files directly to the Cisco Crosswork Planning Collector. External scripts can access these files at run time, without requiring script repackaging or redeployment. This enables scripts in the Cisco Crosswork Planning Collector to use up-to-date, user-uploaded files during execution, supporting more efficient customization.

The key components involved in the process are:

- Data file uploader: Uploads updated data files to the collector.
- Upload directory: Directory where the data file is uploaded.
- External script: Reads the uploaded data files during execution.

Workflow

These are the stages of accessing dynamic data files through external scripts.

1. Upload the updated data file to the Cisco Crosswork Planning Collector's upload directory using the REST API (https://{{server-ip:port}}/cp/collection-service/api/v1/file-gateway).
2. Run the external script, specifying the path to the data file in the upload directory as a command-line argument. argv[6] represents the path of the upload directory. For details on running an external script, see Run an external script against a network model, on page 42.

Note Uploading the same file multiple times overwrites the existing one.

3. The script reads the latest version of the data file at run time, ensuring that it processes current data.

Sample script for updating interface descriptions

This sample Python script, read-from-excel.py, appends a description to every interface in the network with "My IGP metric is <*value*>" using data from the Excel file, description.xlsx.

Script content

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

src = sys.argv[1]
dest = sys.argv[2]
home = sys.argv[5]
srcNet = Network(src)
excel_file = os.path.join(home, "description.xlsx")
wb = openpyxl.load_workbook(excel_file)
sheet = wb.active

row_count = 1
for node in srcNet.model.nodes:
    for iface in node.interfaces:
        cell_obj = sheet.cell(row=row_count, column=1)
        iface.description = 'My IGP metric is ' + str(cell_obj.value)
        row_count = row_count + 1
        print(iface.description)

srcNet.write(dest)
```

How data is collected from third-party devices

Summary

The support modules are executable programs that take arguments to determine what and how to collect. The collected data is used to augment the given plan file and produce an output plan file.

The support modules must

- include capabilities for data collection, such as SNMP
- allow for the input of an auth file to authenticate access to the devices it will poll
- allow for the input of a network access configuration file to manage specifics of collection, such as timeouts, retries, maximum request per device, and so on
- read and write to a plan file

All these constitute the support module framework, which is provided as a template of Python libraries, simplifying the process of data collection from third-party devices.

Workflow

These stages describe how data is collected from third-party devices using support modules.

1. Once the support modules are written, integrate them into the collectors using support module configurations.

2. Specify the type of support module, executable script. The simplest is to use an executable that can be written in Python). Then, provide the script's path.
3. The collectors reorganize the execution to run the support module.

Collectors with support module configurations

Third-party support module configurations are available in these collectors:

- IGP database (Nodes and Interfaces)
- SR-PCE (Nodes and Interfaces)
- LSP
- BGP
- VPN
- Multicast (all the collectors)

Collect data from third-party devices

This topic describes how to collect data from third-party devices using the support module.

Before you begin

- Ensure you have the required support module.
- Complete the steps mentioned in Preconfiguration workflow.

Procedure

Step 1 Decide whether to create a new collection or edit an existing one. For details, see Configure a collection or Edit a collection.

Step 2 To use an external script as a first step of the collection configuration chain, select the startup script option. When using a startup script, you can either skip the basic topology collector or configure it to use the startup script as its source. If you do not use a startup script, you must select one of the basic topology collectors as needed.

Step 3 In the **Advanced modeling** section, select any of the required collectors listed in Collectors with support module configurations, on page 46. Then, click **Next**.

Step 4 In the **Selected collectors** pane on the left, choose the collectors you selected in Step 3 and make all the necessary configuration changes. For more information, refer to the appropriate collector topics.

Note
Ensure that the basic topology parameters are updated to meet your needs. Update the parameters if necessary.

Step 5 To collect data from third-party devices:

a) Check the **Enabled** check box next to the **3rd party support module** parameter.

All support module configuration options appear.

b) Enter the details for these parameters:

- **Execute using**: Select the language you want to use to run the support module. The valid languages are PYTHON, SHELL, and PERL.
- **Executable script**: Enter the full path of the start-up script. This file includes the options for retrieving the start-up script name in the support module file.

  Note
  Ensure you provide the full path of the script. For example, features/src/supportmodule.py. Use only forward slashes (/) in the path and do not start the path with "./" or "/".

- **Support module**: Click **Browse** and select the support module. Ensure that the support module is in .zip or .tar format.

c) (Optional) In the **Optional Arguments** section, enter the relevant arguments as key-value pairs. This is required if you want to collect data from devices based on a specific configuration parameter in the support module.

Step 6 After entering the required configuration parameters in all the selected collectors, click **Next**.

Step 7 Preview the configuration and then click **Create** to create the collection.

Step 8 Configure schedules for the collection job. You can schedule the collection job to run immediately or at specific intervals. For details, see Schedule a collection.

The system begins collecting data from the specified third-party devices using your provided support module and parameters.

Merge AS plan files

This topic describes how to configure the **Merge AS** tool to merge plan files from different Autonomous Systems (ASes).

This tool resolves any conflicts across plan files. It supports plan files in native format.

Important notes on the Merge AS tool

- Each AS can be on a different Cisco Crosswork Planning server.
- Only AS, Circuits, Nodes, Interfaces, External Endpoints, and External Endpoint Members with virtual nodes and unresolved interfaces are resolved.
- These demands are resolved:
  - Source or Destination associated with a virtual node that is resolved with a real node
  - Source or Destination associated with the interface in a specific format
  - Source or Destination associated with the External Endpoints
- These demands are not resolved:
  - Source or Destination associated with ASN number only
- For a given plan file, the internal ASN must match what other plan files identify as an external ASN, and all ASes to be merged must be discovered in every plan file.

Before you begin

- Complete the steps mentioned in Preconfiguration workflow.
- Collect topology and traffic information for different ASes.
- Ensure that the plan files from different ASes are present on the same Cisco Crosswork Planning server, and their file paths are specified.

Procedure

Step 1 Decide whether to create a new collection or edit an existing one. For details, see Configure a collection or Edit a collection.

Step 2 Click the **Tools** radio button at the top.

Step 3 Select **Merge AS** and click **Next**.

Step 4 Enter these configuration parameters:

- **Retain demands**: Check the **Enabled** check box to merge the demands.
- **Tag name**: Enter a tag name to help identify the updated rows in the .pln file. The tag column in the .pln file gets updated with this tag name for modified rows.

Step 5 In the **Source collector** section, click **+ Add source collector**, and select the relevant Collection and Collector names.

Step 6 In the **Source DB** section, click **+ Add source DB**, click **Browse**, and select the source plan file located on your system.

Note
If you are migrating the configuration either from Cisco WAE or from another Cisco Crosswork Planning instance, ensure that the **DB file** field is updated with the correct file after importing the configuration. This is necessary because, after importing the configuration, the server restores only the file name and not the actual file. If the field is not updated with the correct file, then the collection fails.

Step 7 Click **Next**.

Step 8 Preview the configuration and then click **Create** to create the collection.

Step 9 Configure schedules for the collection job. You can schedule the collection job to run immediately or at specific intervals. For details, see Schedule a collection.

The resulting plan file consolidates data from different ASes.

What to do next

Use this collection as the source network to configure additional collections. For details on configuring different collectors, see the relevant topics in this chapter. For details on editing the collection, see Edit a collection.

Representative plan files

A representative plan file is a network plan that

- provides a view that better represents the general network state
- is generated using multiple snapshots from an archive, and

- includes multiple traffic levels, each corresponding to a specified time interval.

Plan files are snapshots of a network state at a point in time. Transitory events, such as failures, are captured in the snapshot if they occur during the collection window. The traffic collected is the specific traffic that occurred during that collection window, which can be five minutes or less. As such, a snapshot usually does not represent the network state over the course of a typical day or week of network operation, and thus, is inadequate to use as the basis for long-term design and planning tasks. To address these needs, the **Create representative plan** tool uses multiple snapshots from an archive to construct a single plan that is more representative of the general network state.

You can use an external script to create a representative plan. Use the **argv[7]** argument in your script to access the archive root directory where all archives are available.

A representative plan file is particularly useful when planning networks where peak utilization occurs at unknown or different times of the day across different interfaces. A simulation analysis performed over all traffic levels identifies the time intervals when peaks occur.

Features of a representative plan

- The topology is extracted from a base plan, which by default is the most recent snapshot. However, you can specify any plan file as the base plan.
- The representative plan contains multiple traffic levels, one for each specified time interval. For example, there could be one traffic level per hour.
- Demands are extracted from snapshots that are selected from each of these time intervals. A single demand in the representative plan file contains a range of traffic values that represent the different amounts of traffic for that demand over the course of the specified time period.
- Interface, LSP, and node measurements are extracted from snapshots that are selected from each of these time intervals.

How the Create representative plan tool works

Summary

The **Create representative plan** tool creates a single plan file from snapshot plans in the archive. It creates traffic levels in the plan, each representative of demand traffic in a given time interval during a day or week.

Workflow

These stages describe how the tool creates a representative plan:

1. The tool examines all snapshots that fall into the specified time interval during the sample time range. Of these snapshots, Cisco Crosswork Planning selects the one with the least number of failed circuits and the most number of active interfaces in the common base plan. If there is a tie, the snapshot with the highest amount of demand traffic is selected. Only snapshots with single traffic levels are used.
2. The tool removes all traffic intervals from the base plan.
3. The tool creates new traffic intervals in the base plan, using the format HHmm-HHmm or dddHHmm-dddHHmm, depending on whether the time period is a day or a week.

   For example, 03:00-04:00 and Fri17:00-Fri18:00.

4. The tool imports all demands from the corresponding snapshot for the time interval.
   - If a snapshot demand matches the one existing in the base plan, then Cisco Crosswork Planning uses that demand and the snapshot demand traffic for it.
   - If there is no matching demand, Cisco Crosswork Planning Design creates the demand with 0 traffic.
   - If a demands exists in the base plan, but not in the snapshot, the demand is used with 0 traffic.
   - The tool also imports multicast demands with the required multicast flows and multicast destinations.
5. The tool imports measured traffic for interfaces, LSPs, and nodes that exist in both the base plan and the snapshot.

Result

Each resulting representative plan file includes a report section where each traffic level is defined per row.

Create a representative plan using the tool

This topic describes how to generate a network plan file that represents typical network conditions using multiple archived network snapshots.

Use this task when you need a plan file that reflects the overall network state, rather than a single moment in time. This is helpful for long-term network design and planning.

Before you begin

- Complete the steps mentioned in Preconfiguration workflow.
- Ensure you have the archived plan files.

Procedure

Step 1 Decide whether to create a new collection or edit an existing one. For details, see Configure a collection or Edit a collection.

Step 2 Click **Tools** at the top of the collection page.

Step 3 Select **Create representative plan** and click **Next**.

Step 4 From the **Archive** drop-down, select the archive from which you want to create the representative plan. The tool uses the snapshots from this archive and generates the final representative plan.

Step 5 Enter the relevant configuration parameters. For descriptions of these parameters, see Representative plan configuration parameters, on page 51.

Step 6 Preview the configuration and then click **Create** to create the collection.

Step 7 Configure schedules for the collection job. You can schedule the collection job to run immediately or at specific intervals. For details, see Schedule a collection.

A representative plan file is generated based on the parameters you configured.

What to do next

You can:

- Access the resultant plan file from the Cisco Crosswork Planning Design application. For details, see View or download plan files.
- View the report from the Cisco Crosswork Planning Design application. From the visualization toolbar, select **Actions** > **Reports** > **Generated reports**. Click the **Representative Plan** link to view the report.

Representative plan configuration parameters

You can fine-tune the results to include specified snapshots for multiple traffic levels across specific intervals. This topic describes the representative plan configuration parameters.

Time interval parameters

These parameters define time intervals during a day or during a week (defined by **Time period**) that are created for the resulting plan file.

Parameter	Description
Time period	Defines whether the time intervals are for a day or a week.
Time interval length	Specifies the length of time intervals for each traffic level, in minutes. The default is 60 minutes. For time period "Day", the maximum limit is 60*24=1,440 minutes. For time period "Week", the maximum limit is 60*24*7=10,080 minutes.
Time interval starts	Defines the starting times for time intervals in the UTC time period. For example, 1600 for 4 PM, Mon1600 for Monday 4 PM. For a time period of one day, the format is "HHmm". For a time period of one week, the format is "dddHHmm". For example, 1600 for 4 PM, Mon1600 for Monday 4 PM. The default is all time intervals in the period, starting at "0000" (day) or "Mon0000" (week).

Sample time range parameters

These parameters define which periods of data in the archive are used to populate the specified time intervals.

Parameter	Description
Sample time end	Specifies the end time for the sample. The format is YYYYMMDD_HHmm. The default is the last inserted date in the archive.
Sample time length	Specifies the length of sample, in days. The default is 1 for time period "Day" and 7 for time period "Week".

Other parameters

These parameters define which periods of data in the archive are used to populate the specified time intervals.

Parameter	Description
Archive	The archive from which you want to create the representative plan. The tool uses the snapshots from this archive and generates the final representative plan.
Archive base time	Specifies the snapshot in the archive at this time as the base plan to augment with traffic levels from other snapshots in the archive. The format is YYYYMMDD_HHmm). Default is the most recent snapshot in the sample period.
Base plan	Specifies the plan file to augment with traffic levels from the archive. If provided, this will override the value specified in **Archive base time**.
Verbosity	Sets the level of detail in log messages. The default is 30, with a valid range from 1 to 60.

Sample parameters and outputs of representative plans

This topic provides sample parameters and outputs of representative plans.

Example 1

In this example, the network peak time is between 4 PM and 7 PM daily. To better understand this peak traffic, we could create a representative traffic level for each hour within this range: 4 PM, 5 PM, and 6 PM. For weekly forecast purposes, we use samples from the last five days to construct the traffic levels. We use the latest snapshot in the time period, 110502_0347_UTC.pln, as the base plan. This plan is located in the archive named “backbone”.

Parameters used:

- Archive: backbone
- Base plan: 110502_0347_UTC.pln
- Time interval length: 60
- Sample time length: 1
- Time interval starts: 1600, 1700, 1800

Output:

Traffic level	Snapshot	Matching interfaces	Total demand traffic	Total demands	Demands not imported
16:00-17:00	110502_0347_UTC.pln	25	43534.32	453	4
17:00-18:00	110502_0347_UTC.pln	25	47583.23	454	3
18:00-19:00	110502_0347_UTC.pln	25	50771.49	454	3

Example 2

In this example, we know the network peak times are around 4 PM daily, as well as 8 PM on Fridays. We need to get a representative traffic level for each of these six periods for the last two weeks. Because today is Tuesday, yesterday's 4-5 PM range and last week's Monday 4-5 PM range are used to construct the 4 PM traffic level. The base plan, 110407_0423_UTC.pln, is in the "acme" directory.

Parameters used:

- Base plan: 110407_0423_UTC.pln
- Time period: week
- Time interval length: 60
- Sample time length: 14
- Time interval starts: Mon1600, Tue1600, Wed1600, Thu1600, Fri1600, Fri2000

Output:

Traffic level	Snapshot	Matching interfaces	Total demand traffic	Total demands	Demands not imported
Mon16:00-Mon17:00	110407_0423_UTC.pln	97	43534.32	6702	21
Tue16:00-Tue17:00	110407_0423_UTC.pln	97	47583.23	6702	21
Wed16:00-Wed17:00	110407_0423_UTC.pln	95	50771.49	6701	22
Thu16:00-Thu17:00	110407_0423_UTC.pln	97	56831.91	6702	21
Fri16:00-Fri17:00	110407_0423_UTC.pln	93	48732.18	6700	23
Fri120:00-Fri21:00	110407_0423_UTC.pln	97	53692.39	6702	21