NetBrain® Integrated Edition 10.1

Release Note

Table of Contents

1 Key New Features and Enhancements

NetBrain IE v10.1 introduces the following new features and enhancements:

1. Problem Diagnosis Automation System (PDAS)

IE v10.1 greatly enhances the **Problem Diagnosis Automation System (PDAs)**, which automates the Diagnosis of repetitive problems and enforces preventive measures across the entire network with the following new features and enhancements:

- Network Intent Cluster (NIC) New: NIC clones a Network Intent (NI), seed NI, across the entire network to create a group of NIs (member NIs) with the same design or logic. NIC can be created from the seed NI via the 7-step, no coding process. In PDAS, a subset of Member NIs can be automatically executed according to the user-defined condition based on the member device, the member NI tags, or signature variables.
- Triggered Automation Framework (TAF) New: TAF matches the incoming API calls from the 3rd IT system such as ServiceNow to NetBrain Incidents and installs the automation (NI/NIC) to be triggered for each call. It has three key components: Integrated IT System defining the scope and data of the incoming API calls, Incident Type to match a call to a NetBrain Incident, Triggered Diagnosis to define what and how the NIC/NI is executed.
- ServiceNow App 3.0: the new version provides more flexible control on what data will be sent to NetBrain IE and supports multi-tenant deployment. It also integrates the NetBrain Incident pane into ServiceNow instead of the static map in the earlier versions.
- Incident Pane Enhancements: as the output of PDAS, Incident Pane of IEv10.1 provides richer data and diagnosis history, including NI diagnosis results (from TAF, Probe, manually run), the status codes of Network Intent, diagnosis summary message, and recommended diagnoses.
- Path-Based Troubleshooting Flow (PBTF): IE v10.1 introduces the Path Intent and many enhancements to enable users to baseline, document, and define the diagnosis logic for application paths when the network is healthy. Then this well-documented and executable knowledge can help network engineers resolve application slowness issues more efficiently or do application impact analysis when a problem occurs on a network device or device interface.
- Parser Discovery New: a new feature to enable the user to find a Parser by the parser name, description, CLI command, variable name, and keywords of the sample text and test it with the local device data.
- Visual Parser Improvements: support SNMP as the data source and add a new parser group type: Collector.
- NI Improvements: support compound tables from different devices, formula columns to preprocess the table column data, macro variable, CSV output, and run NI with current baseline data. IE v10.1 also

introduces the NI manager and adds two helpful functions to NI Editor, the Duplicate device section and switch devices.

2. Enhance the data accuracy and self-maintenance

The other focus of IE v10.1 is to enhance the data accuracy and system self-maintenance with the following new features and enhancements:

- Open Topology (OT) New: a new framework to calculate the L3 and L2 topology. Besides redesigning the algorithm to improve the topology accuracy, OT is an open process to incorporate the user's input in every stage of the process and is easier to maintain. OT takes the data from the network (Interface, MAC Table, ARP Table, CDP/LLDP, routing neighbors, etc.) and the user input to calculate the VLAN Group and then build L3 and L2 topology inside each VLAN Group.

- Platform Validation (PV) New: PV checks the data accuracy by executing a set of rules called PV Rule (PVR). Besides the validation logic, a PVR will create a unique error code, error message, and recommended actions. A scheduled PV task will expose the data accuracy once, and the results are displayed in the Data Accuracy Wizard (DAW) per device.

- System Validation (SV) New: SV collects all data accuracy issues caused by the system misconfiguration and live access issues and displays the results in DAW with the error code, message, and recommendations.

- Data Accuracy Wizard (DAW) New: DAW is central to addressing the data accuracy issues per device. Users can view all info related to the data accuracy, including all data errors reported by PV and SV with the recommended actions, the topology and VLAN groups created by OT, PV Rules, and Open Drivers applicable to this device.

- Open Driver (OD) New: OD provides an open platform for the user to fix the data accuracy issues in a device or a group of devices. It defines the logic (target data, data source, and mapping of the parser variables to the target date) to set empty data or modify an incorrect one.

3. New Foundation Modules

- Google Cloud Visual Management New: the support of Google Cloud Platform (GCP) is added in v10.1, including the discovery of GCP resources, mapping the GCP objects, mapping the application dependency across GCP, SPOG access through GCP native and 3rd party cloud management tools, etc.

- IPv6 Network Support New: add the support of the single-stack IPv6 network, including the discovery of IPv6 only network devices, L3/L2 topology, and mapping.

4. **Collaborative Troubleshooting Enhancements**

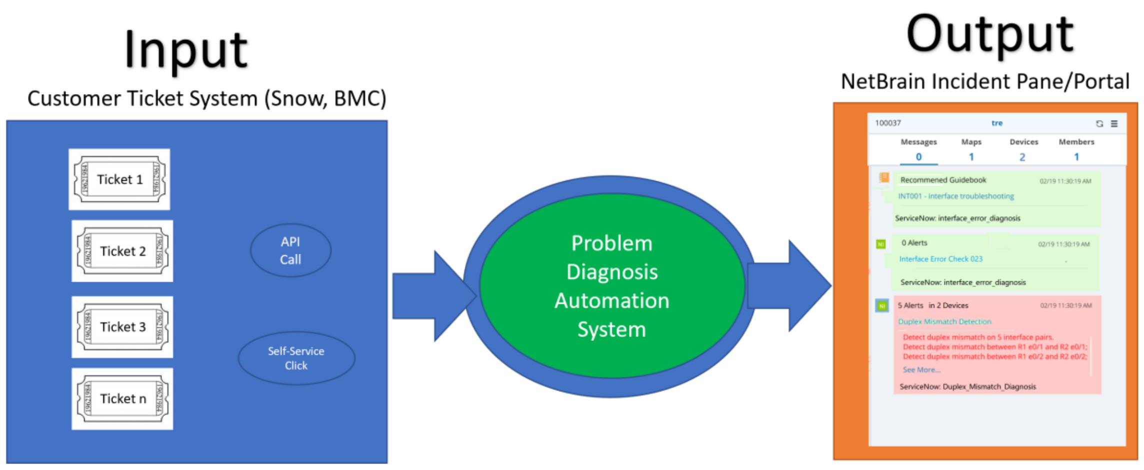
- Personal Map Copy New: users can create a personal copy of a shared map (the master map) and share their findings and changes with others without altering the master map.
- Reference Map Enhancements: provides a dialog to browse and select all common maps (standalone maps) and function maps (site, device group, Intent, Path, etc.) and allows a user to choose any common and function map as the reference map for the Path, Intent, etc.
- Site Map Enhancements: support adding more devices like linked neighbor devices to a site map.
- Smart CLI: support the Smart CLI on MAC OS.

5. **Other Enhancements**

- KC and Auto Update Enhancements: enable the silent downloading and installation of the platform resources, etc.
- License Enhancements: support two license modes, separate pool and universal poll, and usage-based license model for IBA module.
- Benchmark, Live Access, and Fine-Tune Enhancements: enhancements to the device log of benchmark and fine-tune UI. Allow the users to lock only one setting, such as Management IP, in the device setting.
- Installation Pre-check Tool New: develop a tool to check users' system readiness to install/upgrade and generate a report to help users prepare appropriately for the installation/upgrade.
- Search Enhancements: can search automation objects, NAT table, and Virtual server table.
- Other Enhancements and Adjustments

2 Problem Diagnosis Automation System (PDAS)

Release v10.1 greatly enhances the **Problem Diagnosis Automation System (PDAs)**, which automates the Diagnosis of repetitive problems and enforces preventive measures across the entire network. As illustrated in the following diagram, from the end user's perspective, the output of PDAs is NetBrain Incident Pane/Portal, a central collaboration platform for troubleshooting and data sharing for each problem.



Recommened Guidebook 02/19 11:30:19 AM
INT001 - interface troubleshooting
ServiceNow: interface_error_diagnosis

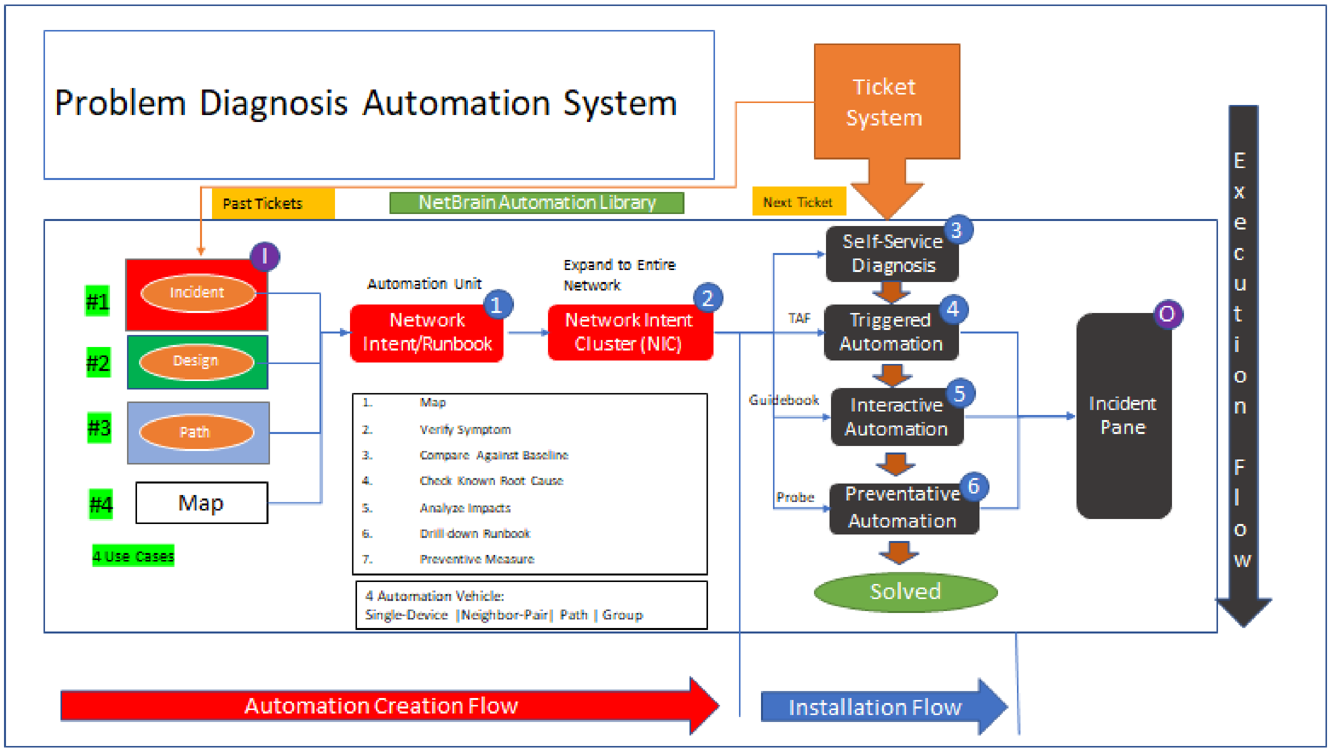
0 Alerts 02/19 11:30:19 AM
Interface Error Check 023
ServiceNow: interface_error_diagnosis

5 Alerts in 2 Devices 02/19 11:30:19 AM
Duplex Mismatch Detection
Detect duplex mismatch on 5 interface pairs.
Detect duplex mismatch between R1 e0/1 and R2 e0/1;
Detect duplex mismatch between R1 e0/2 and R2 e0/2;
See More...
ServiceNow: Duplex_Mismatch_Diagnosis

Input and Output of Problem Diagnosis Automation System (PDAS)

The underlying system has three essential flows, as shown in the following system architecture diagram:

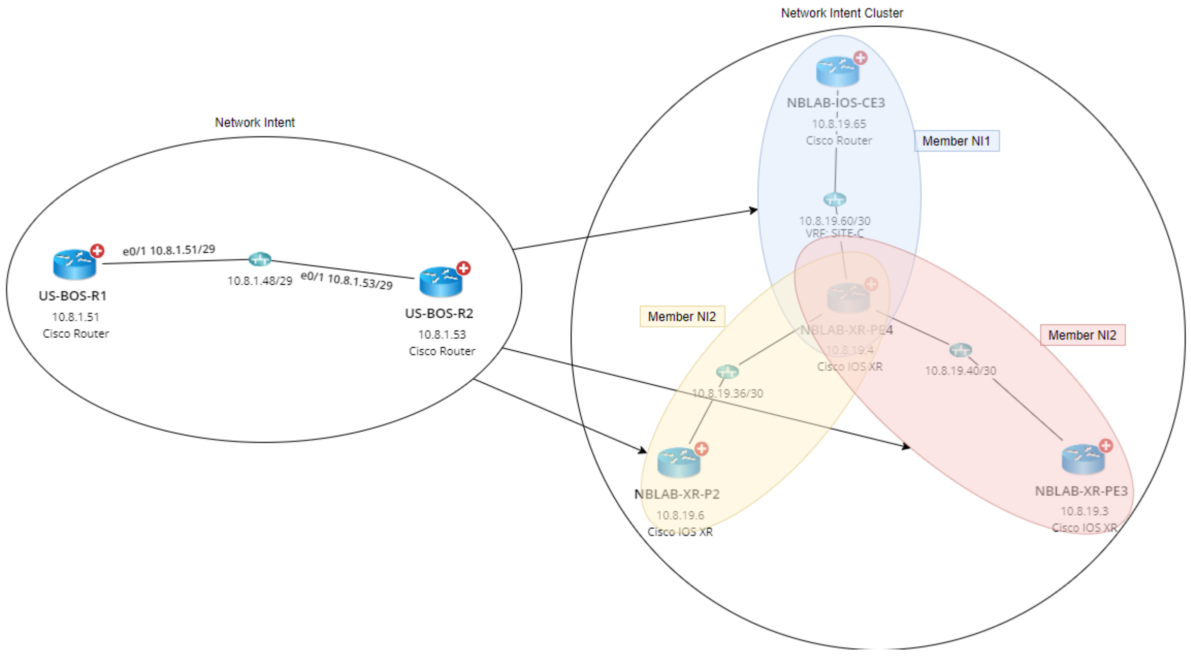
- **Automation Creation Flow**: where diagnosis know-how is turned into automation assets across the entire network in the form of Network Intent (NI) or Executable Runbook (RB) inside the no-code platform.
- **Automation Installation Flow**: where various automation assets are connected to future problem diagnosis through Trigger from the ticket system, or human interaction, or NetBrain's adaptive monitoring system.
- **Automation Execution Flow** – where automation is executed in response to an external symptom in three successive methods, namely triggered, interactive, and preventive. All execution output is organized inside the NetBrain incident pane for each distinctive Incident.



2.1 Network Intent Cluster (NIC)

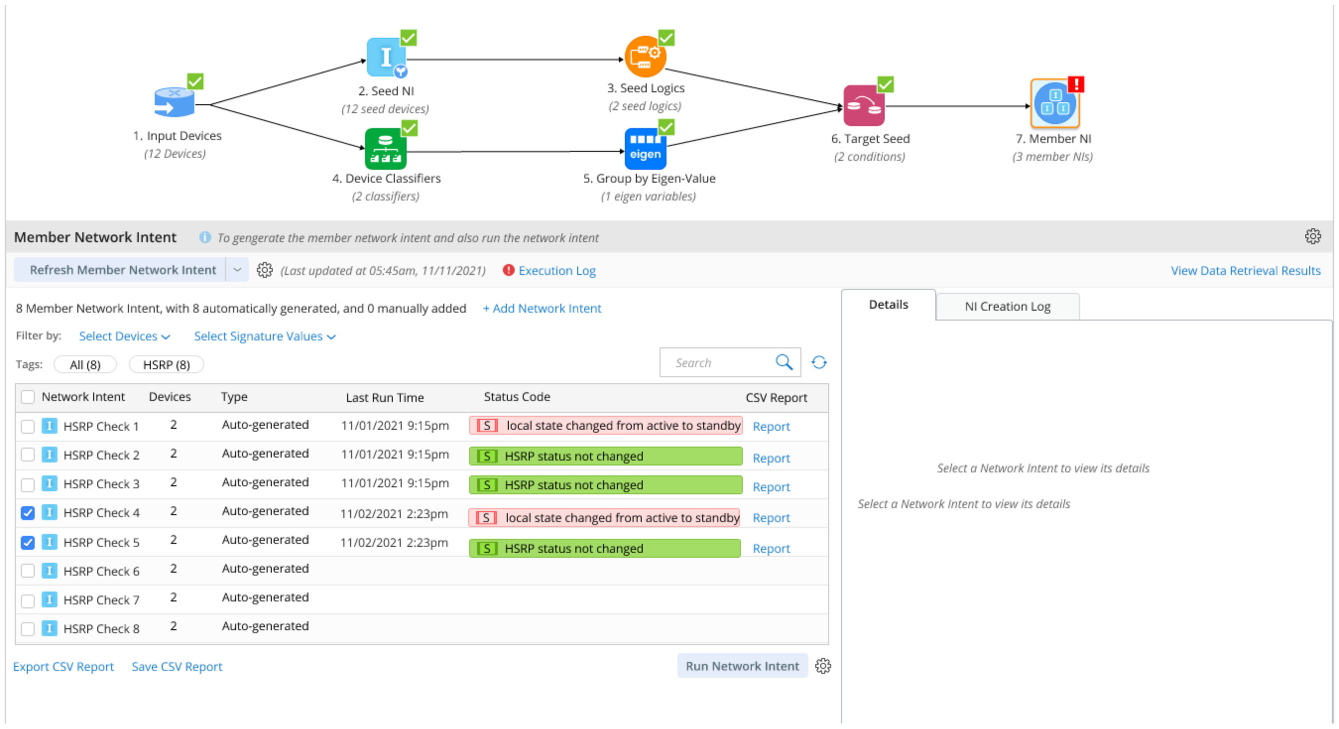
V10.1 introduces NIC, which expands Network Intent (NI) scope from a specific network design to one type of network design with similar diagnosis logic. While NI effectively documents and validates a network design, it applies to only one network device or a set of devices at a time. Therefore, it can take many repetitive efforts to create NIs for a large network. NIC is designed to expand the logic of a NI (seed NI) from one or a set of devices to the whole network. Furthermore, NIC can be triggered to run in the Triggered Automation Framework (TAF), and its results can significantly reduce the MTTR. NIC requires no coding skills and has an intuitive user interface for creating and debugging.

For example, you create a NI to monitor whether failover occurs between a pair of HRSP devices, ***US-BOS–R1*** and ***US-BOS-R2*** (the failover often causes the performance issue such as the slow application). Then, NIC can replicate the logic to all pairs of HRSP in the network without any coding.



NIC is composed of a group of NIs (Member NIs) cloned from Seed NI via a 7-step, no-code process. A NIC may have thousands of Member NIs, corresponding to a specific network diagnosis. A subset of Member NIs can be selected to execute according to the user-defined matching logic based on: (1) devices inside the member NI (member device), 2) unique tag for each Member NI, or 3) signature variables assigned to Member NI.

The following diagram is a sample NIC to clone a seed NI to check the HSRP running status for a network site. By creating a NIC to achieve this, you can expand the Diagnosis of one site to your entire network. Each Member NI has its tag and signature variable, the ***virtual IP address*** of HSRP.



To gengerate the member network intent and also run the network intent

(Last updated at 05:45am, 11/11/2021)

8 Member Network Intent, with 8 automatically generated, and 0 manually added

Network Intent	Devices	Type	Last Run Time	Status Code	CSV Report
HSRP Check 1	2	Auto-generated	11/01/2021 9:15pm	[S] local state changed from active to standby	Report
HSRP Check 2	2	Auto-generated	11/01/2021 9:15pm	[S] HSRP status not changed	Report
HSRP Check 3	2	Auto-generated	11/01/2021 9:15pm	[S] HSRP status not changed	Report
✓ HSRP Check 4	2	Auto-generated	11/02/2021 2:23pm	[S] local state changed from active to standby	Report
✓ HSRP Check 5	2	Auto-generated	11/02/2021 2:23pm	[S] HSRP status not changed	Report
HSRP Check 6	2	Auto-generated			
HSRP Check 7	2	Auto-generated			
HSRP Check 8	2	Auto-generated			

Select a Network Intent to view its details

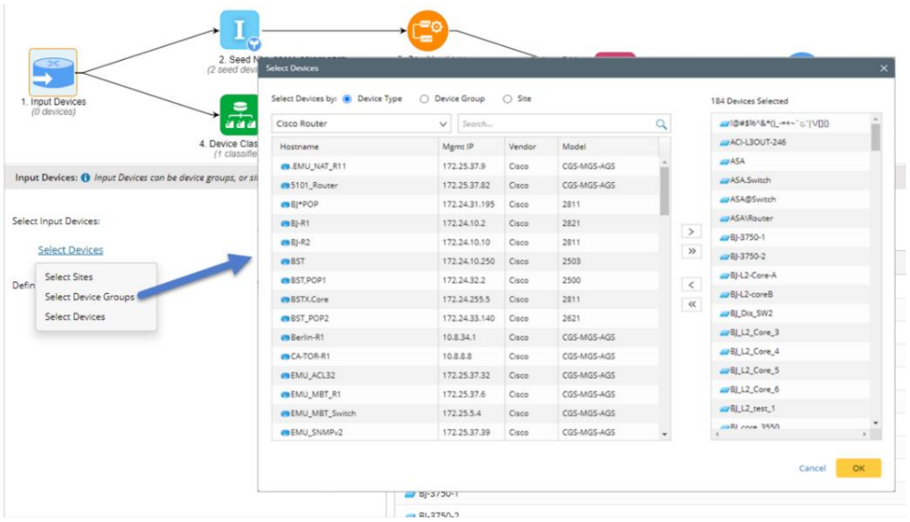
Select a Network Intent to view its details

2.1.1 NIC Creation Flow

V10.1 provides via a 7-step, no-code process to create a NIC.

2.1.1.1 Input Devices

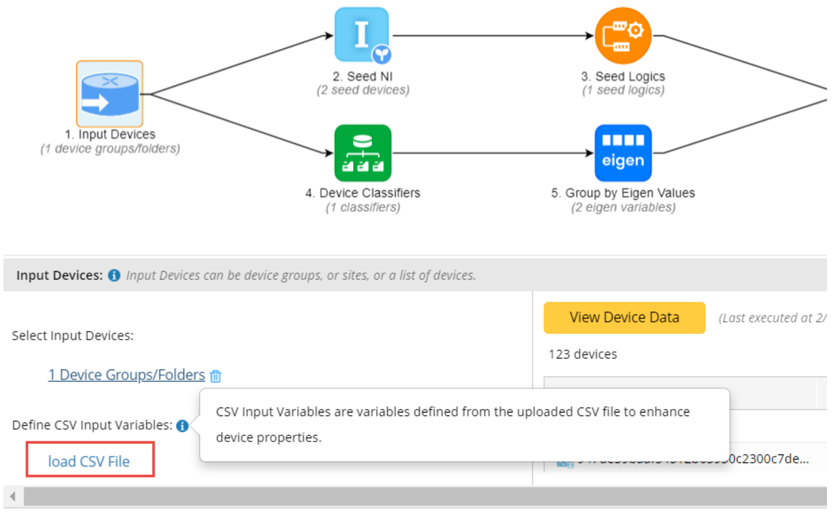
In the **Input Devices** node, you select the devices you want to expand the NI via the site, device group, and individual devices.



Hostname	Mgmt IP	Vendor	Model
.EMU_NAT_R11	172.25.37.9	Cisco	CGS-MGS-AGS
5101_Router	172.25.37.82	Cisco	CGS-MGS-AGS
BJ*POP	172.24.31.195	Cisco	2811
BJ-R1	172.24.10.2	Cisco	2821
BJ-R2	172.24.10.10	Cisco	2811
BST	172.24.10.250	Cisco	2503
BST.POP1	172.24.32.2	Cisco	2500
BSTX.Core	172.24.255.5	Cisco	2811
BST_POP2	172.24.33.140	Cisco	2621
Berlin-R1	10.8.34.1	Cisco	CGS-MGS-AGS
CA-TOR-R1	10.8.8.8	Cisco	CGS-MGS-AGS
EMU_ACL32	172.25.37.32	Cisco	CGS-MGS-AGS
EMU_MBT_R1	172.25.37.6	Cisco	CGS-MGS-AGS
EMU_MBT_Switch	172.25.5.4	Cisco	CGS-MGS-AGS
EMU_SNMPv2	172.25.37.39	Cisco	CGS-MGS-AGS

- !@#$%^&*()_-+=~`;:'|\/[]0
- ACI-L3OUT-246
- ASA
- ASA.Switch
- ASA@Switch
- ASA\Router
- BJ-3750-1
- BJ-3750-2
- BJ-L2-Core-A
- BJ-L2-coreB
- BJ_Dis_SW2
- BJ_L2_Core_3
- BJ_L2_Core_4
- BJ_L2_Core_5
- BJ_L2_Core_6
- BJ_L2_test_1

Users can load a CSV File to import the variables to enhance the device properties (the interface-related data is not supported in v10.1).



Input Devices: Input Devices can be device groups, or sites, or a list of devices.

Select Input Devices:

Define CSV Input Variables:

CSV Input Variables are variables defined from the uploaded CSV file to enhance device properties.

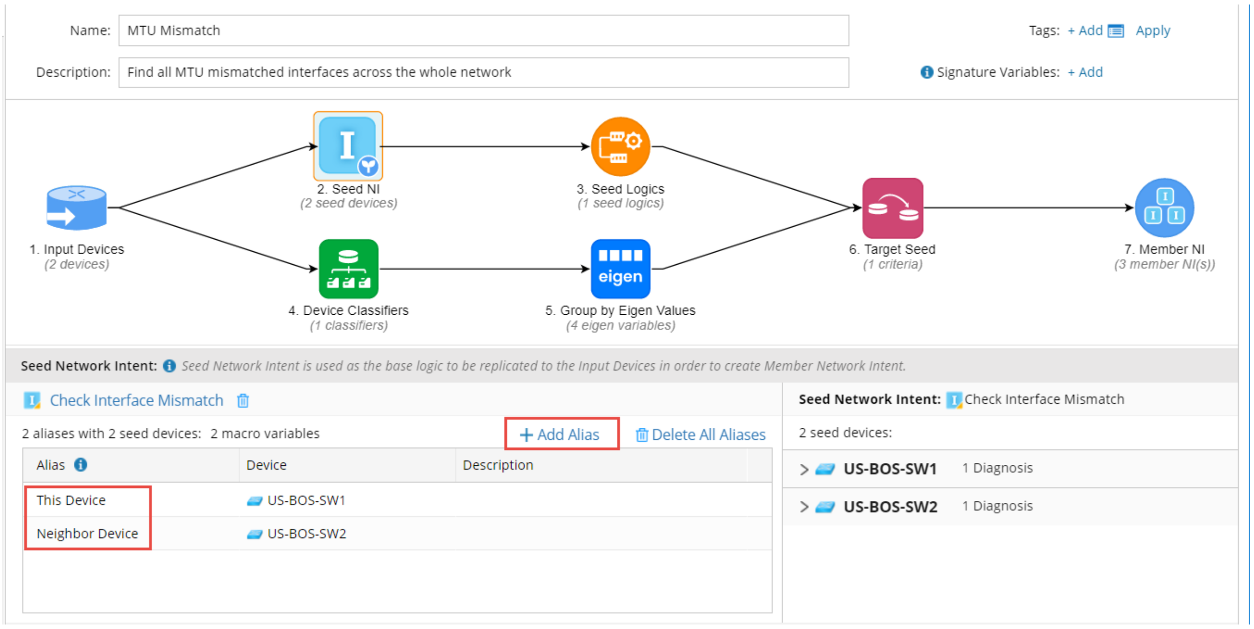
The CSV Input Variables can be used in the following functions:

- **Eigen Variable Identification:** The CSV input variables can be selected and used to define Eigen Variable to divide devices into different Eigen groups for NI creation (step 5).

- **Target Seed Logic**: CSV input variables can be used in the Target Seed condition (step 6).
- **Macro Variable**: You may want to pass the device property to NI via Macro Variable, and you can use the CSV Input variable to achieve this (step 7).

2.1.1.2 Seed NI

Seed NI node selects a NI you want to expand the logic. The seed devices will have default alias, ***D1***, ***D2***, etc. Users can change the alias to an intuitive name, such as ***this device, neighbor device***. Only one NI can be selected for a NIC.



Name: MTU Mismatch

Description: Find all MTU mismatched interfaces across the whole network

Seed Network Intent: Seed Network Intent is used as the base logic to be replicated to the Input Devices in order to create Member Network Intent.

2 aliases with 2 seed devices: 2 macro variables

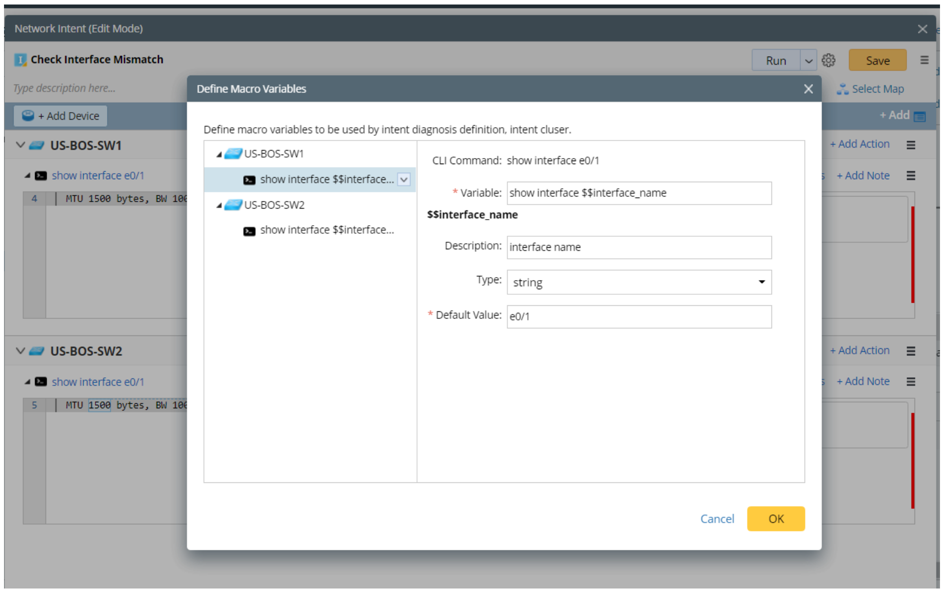
Alias	Device	Description
This Device	US-BOS-SW1	
Neighbor Device	US-BOS-SW2	

Seed Network Intent: Check Interface Mismatch

2 seed devices:

US-BOS-SW1	1 Diagnosis
US-BOS-SW2	1 Diagnosis

The seed NI can support **macro variables**. For example, users can create a NI to check the MTU mismatch between two specific neighbor interfaces using the CLI command ***show interface e0/0***. While replicating this NI to all neighbor interfaces of a network, the system needs to replace the interface name ***e0/0*** with the interface name of the member device. The Macro Variables are defined for this purpose.



Network Intent (Edit Mode)

Check Interface Mismatch

Define Macro Variables

Define macro variables to be used by intent diagnosis definition, intent cluser.

- US-BOS-SW1
  - show interface $$interface...
- US-BOS-SW2
  - show interface $$interface...

CLI Command: show interface e0/1

* Variable: show interface $$interface_name

$$interface_name

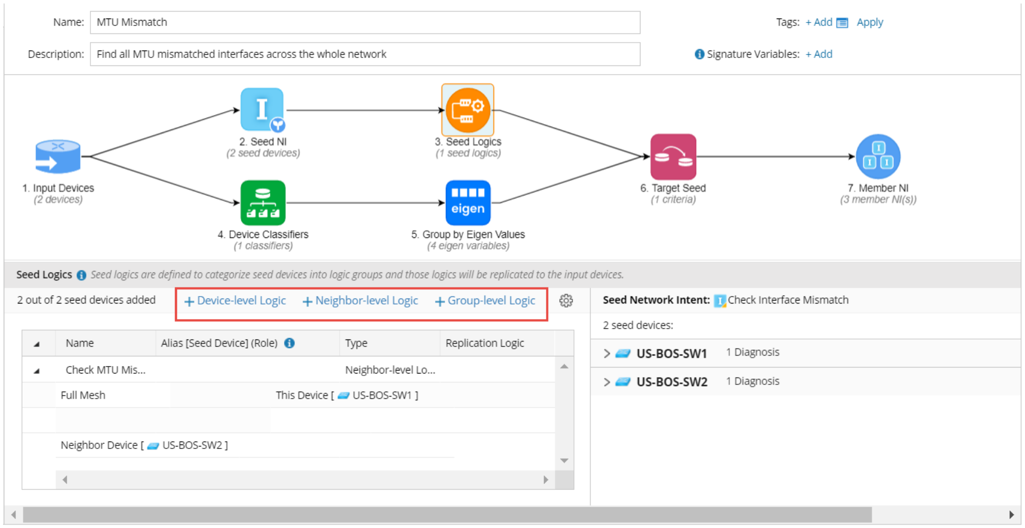
Description: interface name

Type: string

* Default Value: e0/1

2.1.1.3 Seed Logics

The **seed Logic** node selects the logic replicating the seed NI to the input devices. Each seed device can belong to only one seed logic. A seed logic includes a group of seed devices and a replication setting. There are three types of seed logic:



Name: MTU Mismatch

Description: Find all MTU mismatched interfaces across the whole network

1. Input Devices (2 devices)
2. Seed NI (2 seed devices)
3. Seed Logics (1 seed logics)
4. Device Classifiers (1 classifiers)
5. Group by Eigen Values (4 eigen variables)
6. Target Seed (1 criteria)
7. Member NI (3 member NI(s))

Seed Logics *Seed logics are defined to categorize seed devices into logic groups and those logics will be replicated to the input devices.*

2 out of 2 seed devices added | + Device-level Logic | + Neighbor-level Logic | + Group-level Logic

Name	Alias [Seed Device] (Role)	Type	Replication Logic
Check MTU Mis...		Neighbor-level Lo...	
Full Mesh	This Device [US-BOS-SW1]		
Neighbor Device [US-BOS-SW2]			

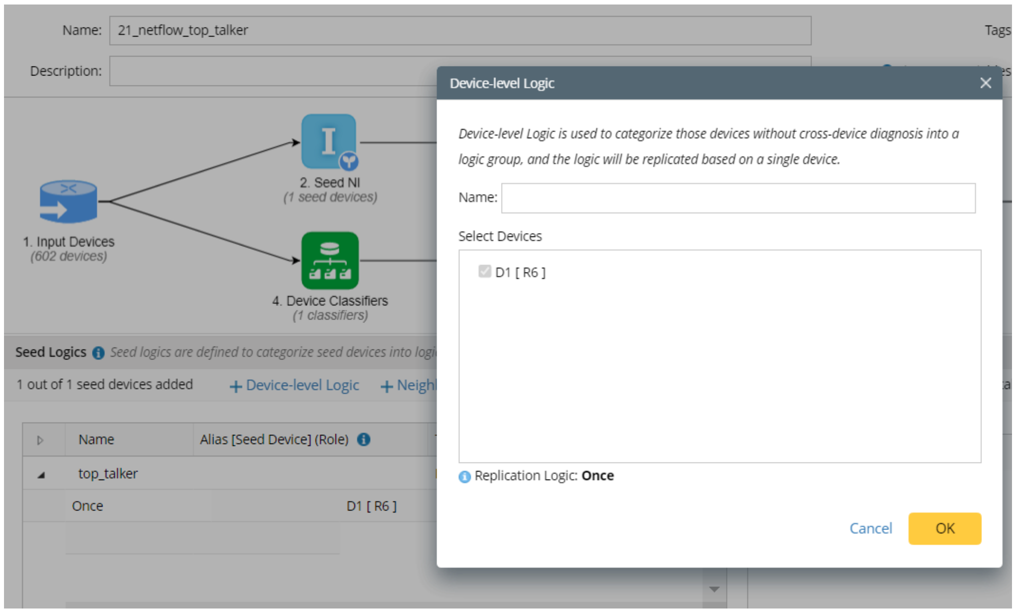
Seed Network Intent: Check Interface Mismatch

2 seed devices:

US-BOS-SW1	1 Diagnosis
US-BOS-SW2	1 Diagnosis

1. Device-level logic

Device-level logic is used for the single device diagnosis and replicated **once** for each device. For example, the NI checks the configurations for security compliance (whether the password is encrypted and telnet is disabled) and monitors the operation status (***interface CRC error*** increases).



Seed logics are defined to categorize seed devices into logi

1 out of 1 seed devices added

top_talker

Once D1 [R6]

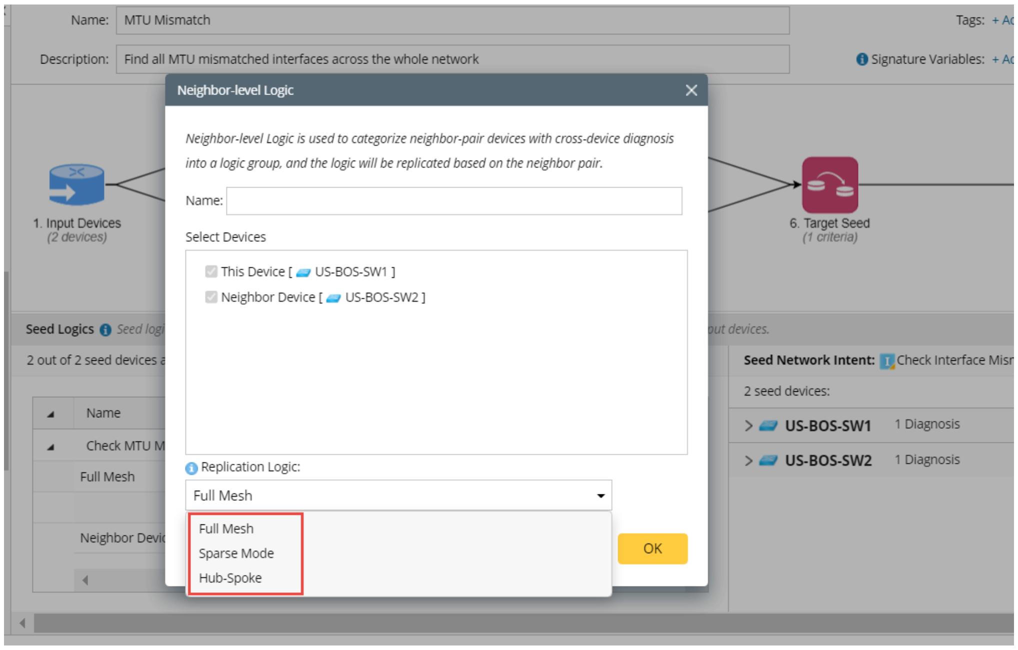
Device-level Logic is used to categorize those devices without cross-device diagnosis into a logic group, and the logic will be replicated based on a single device.

D1 [R6]

Replication Logic: **Once**

2. Neighbor-level logic

Neighbor-level logic is used to categorize neighbor-pair devices with cross-device diagnosis into a logic group, and the logic will be replicated based on the neighbor pair. There are three types of replication logic which is designed for the different types of real-world cases:



Name: MTU Mismatch

Description: Find all MTU mismatched interfaces across the whole network

Neighbor-level Logic

Neighbor-level Logic is used to categorize neighbor-pair devices with cross-device diagnosis into a logic group, and the logic will be replicated based on the neighbor pair.

Name:

Select Devices

- This Device [US-BOS-SW1]
- Neighbor Device [US-BOS-SW2]

Replication Logic: Full Mesh

- Full Mesh
- Sparse Mode
- Hub-Spoke

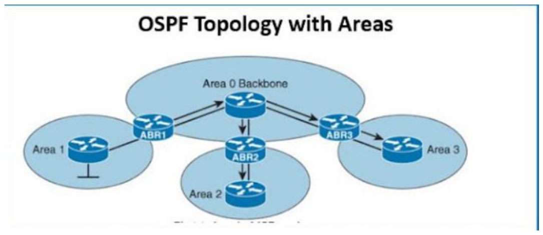
Seed Network Intent: Check Interface Misr

2 seed devices:

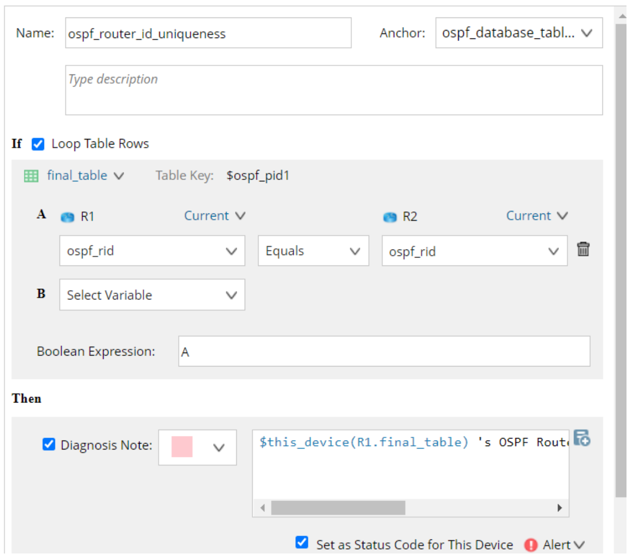
US-BOS-SW1	1 Diagnosis
US-BOS-SW2	1 Diagnosis

a. Full Mesh

The full mesh will take any two input devices in an eigen group to replicate the diagnosis. So, if there are ***n*** input devices in an eigen group, NIC may generate the maximum of ***n*(n-1)/2*** diagnosis in a member NI. Full mesh mode can be used to check the parameters across each neighbor pair to ensure the parameter for each device is unique. For example, check Router IP for an OSPF autonomous system to ensure that all router IDs configured within the same Autonomous system are unique.



Seed NI Logic checks the router ID of two devices to ensure that they are not the same. If the router IDs are the same, the system will raise an alert.



Name: ospf_router_id_uniqueness Anchor: ospf_database_tabl...

Type description

If ☑ Loop Table Rows

final_table Table Key: $ospf_pid1

A R1 Current R2 Current

ospf_rid Equals ospf_rid

B Select Variable

Boolean Expression: A

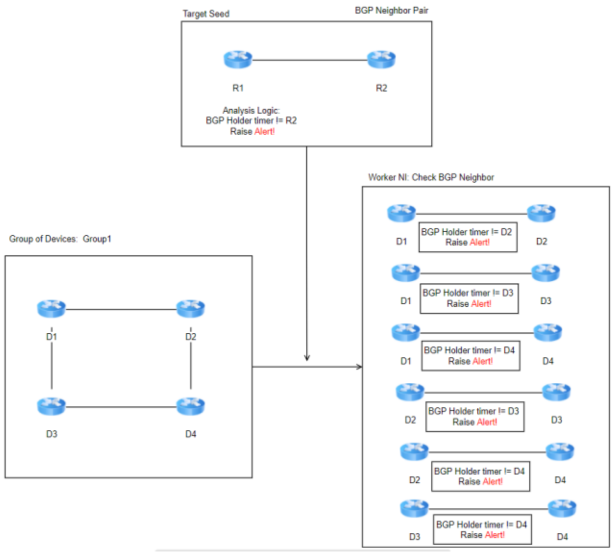
Then

☑ Diagnosis Note:

```
$this_device(R1.final_table) 's OSPF Rout
```

☑ Set as Status Code for This Device Alert

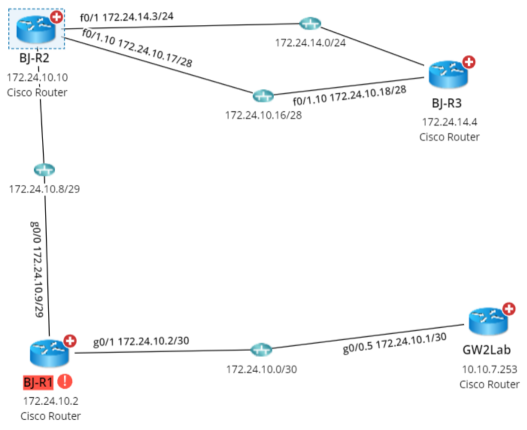
To expand the logic to all devices within the same OSPF autonomous system, use the **Full-mesh** replication logic to define the seed logic.



Target Seed
BGP Neighbor Pair
R1
R2
Analysis Logic:
BGP Holder timer != R2
Raise Alert!
Worker NI: Check BGP Neighbor
Group of Devices: Group1
D1
D2
D3
D4
BGP Holder timer != D2 Raise Alert!
BGP Holder timer != D3 Raise Alert!
BGP Holder timer != D4 Raise Alert!
BGP Holder timer != D3 Raise Alert!
BGP Holder timer != D4 Raise Alert!
BGP Holder timer != D4 Raise Alert!

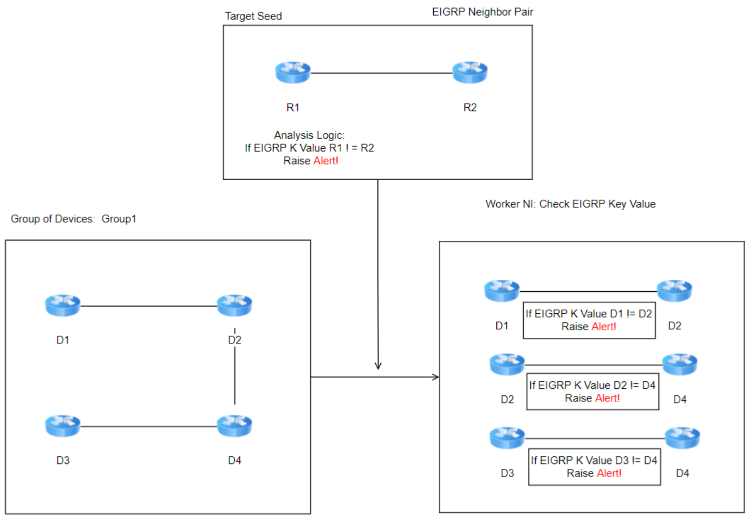
b. Sparse Mode

Sparse Mode will take the input devices of an eigen group as a list and replicate the diagnosis for any two adjacent devices. So, if there are ***n*** input devices, NIC may generate the maximum of ***n-1*** diagnosis in a member NI. Sparse Mode can check the parameters across each neighbor pair to ensure that the parameters are the same across the device selected. For example, *check EIGRP **K** Value for the same EIGRP **AS** number to ensure that all EIGRP key values within the same EIGRP **AS** number are the same*.



Seed NI checks the ***K*** value for two devices to ensure they are not the same. If Key Values are not the same, the system will raise an alert.

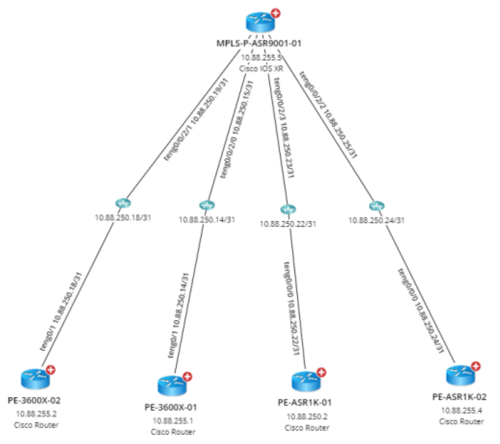
To expand the logic to all devices within the same EIGRP system, we use the **Sparse Mode** replication logic to define the seed logic.


Target Seed
EIGRP Neighbor Pair
R1
R2
Analysis Logic:
If EIGRP K Value R1 ! = R2
Raise Alert!
Group of Devices: Group1
D1
D2
D3
D4
Worker NI: Check EIGRP Key Value
If EIGRP K Value D1 != D2
Raise Alert!
If EIGRP K Value D2 != D4
Raise Alert!
If EIGRP K Value D3 != D4
Raise Alert!

c. Hub-spoke Mode

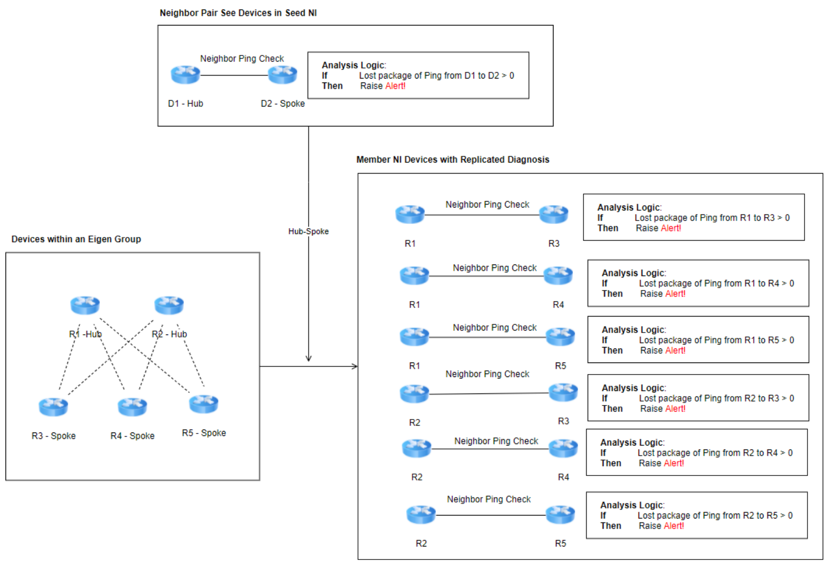
Hub-spoke mode is applied to the pair of devices with different roles. For example, one is a P device, and the other is a PE device. Hub-spoke mode will divide the input devices of an eigen group into two groups according to the roles and take one device from each group to replicate the diagnosis. For example, if there are ***m*** P devices and ***n*** PE devices, NIC may generate the maximum ***m*n*** diagnosis in a member NI (for this eigen group).

We can create a NI to check the connectivity between a P and a PE device to ensure their connectivity is working. Then expand the check logic to all connections between P devices and PE devices with Hub-spoke mode.



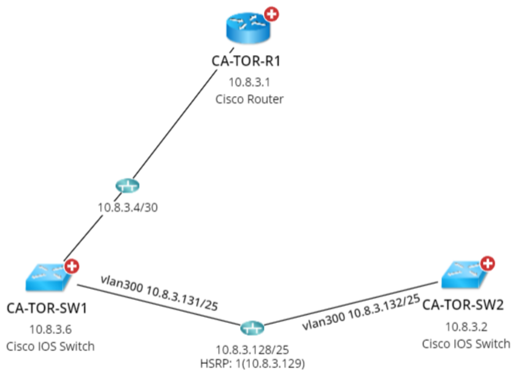
Seed NI checks the connectivity between P and PE devices. If there is a connectivity issue between the P and PE devices, the system will raise an alert.

To expand the logic to all devices within the same BGP AS Number, use the **Hub-Spoke** replication logic to define the seed logic.



3. Group-level logic

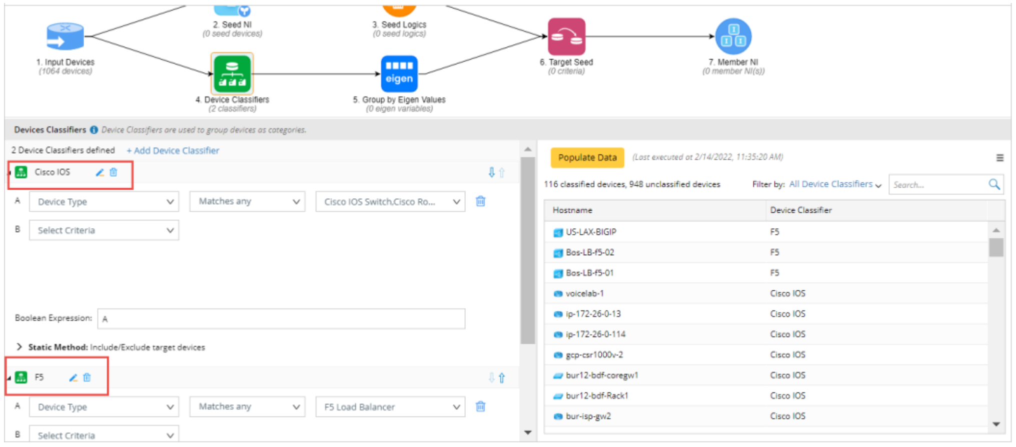
Group Level logic is used to replicate the exact number of device logic with seed NI. For example, a typical remote site of your network consists of one router and two switches.



You create a Seed NI to check the configuration compliance for a particular site, and you want to expand the same logic to all remote sites having the same deployment and setup. You can use **group-level logic** for this purpose.

2.1.1.4 Device Classifier

The Device Classifier node puts devices into different classifiers based on the device types so each classifier can use the same CLI command(s) to retrieve the data or use the same system. Besides device type, users can use other device properties and the configuration file.



Devices Classifiers *Device Classifiers are used to group devices as categories.*

2 Device Classifiers defined + Add Device Classifier

Cisco IOS

A Device Type | Matches any | Cisco IOS Switch,Cisco Ro...

B Select Criteria

Boolean Expression: A

Static Method: Include/Exclude target devices

F5

A Device Type | Matches any | F5 Load Balancer

B Select Criteria

(Last executed at 2/14/2022, 11:35:20 AM)

116 classified devices, 948 unclassified devices

Filter by: All Device Classifiers

Hostname	Device Classifier
US-LAX-BIGIP	F5
Bos-LB-f5-02	F5
Bos-LB-f5-01	F5
voicelab-1	Cisco IOS
ip-172-26-0-13	Cisco IOS
ip-172-26-0-114	Cisco IOS
gcp-csr1000v-2	Cisco IOS
bur12-bdf-coregw1	Cisco IOS
bur12-bdf-Rack1	Cisco IOS
bur-isp-gw2	Cisco IOS

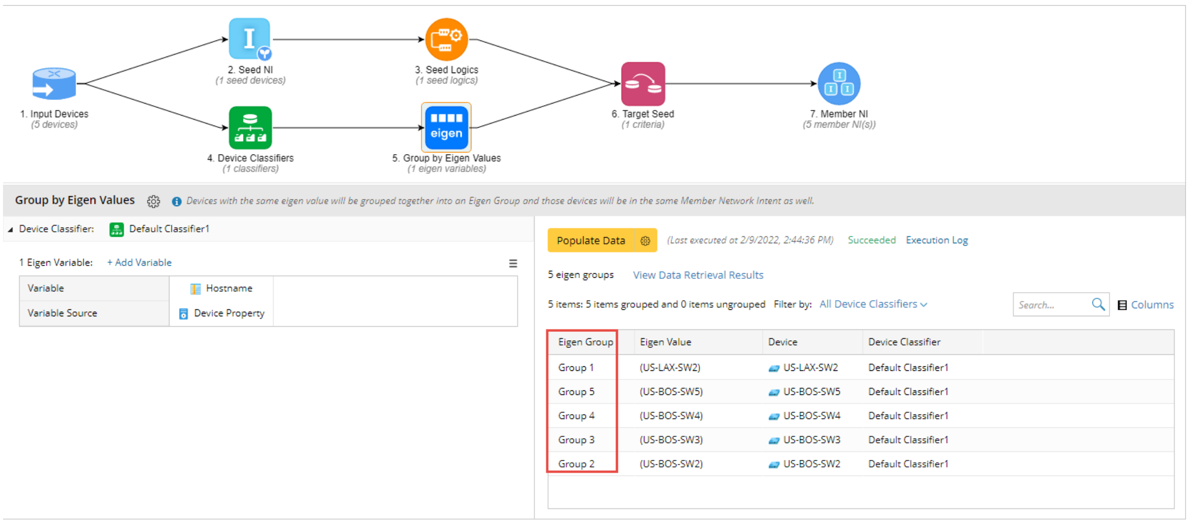
Users can define multiple classifiers, for example, one classifier for one vendor, which can be useful for an NI to support the multi-vendor.

2.1.1.5 Group by Eigen Values

Group by Eigen Value node groups devices with the same eigen value into an **Eigen Group**, which includes three steps:

a. According to the user's selection, the system computes a single string or a list of strings for each device's **eigen value**. For example, the hostname can be the eigen value for a single device diagnosis; for the neighbor level logic with the interface, users may select two neighbor device hostname and two neighbor interface names as the eigen value.

b. The devices with the same eigen value or overlapping eigen value (if a list of strings is used for eigen-value) will form an **eigen group**. Many groups could be formed as a result.
c. The devices in each group will serve as the candidate to form a unique **member NI**.



Devices with the same eigen value will be grouped together into an Eigen Group and those devices will be in the same Member Network Intent as well.

Device Classifier: Default Classifier1

1 Eigen Variable:

Variable	Hostname
Variable Source	Device Property

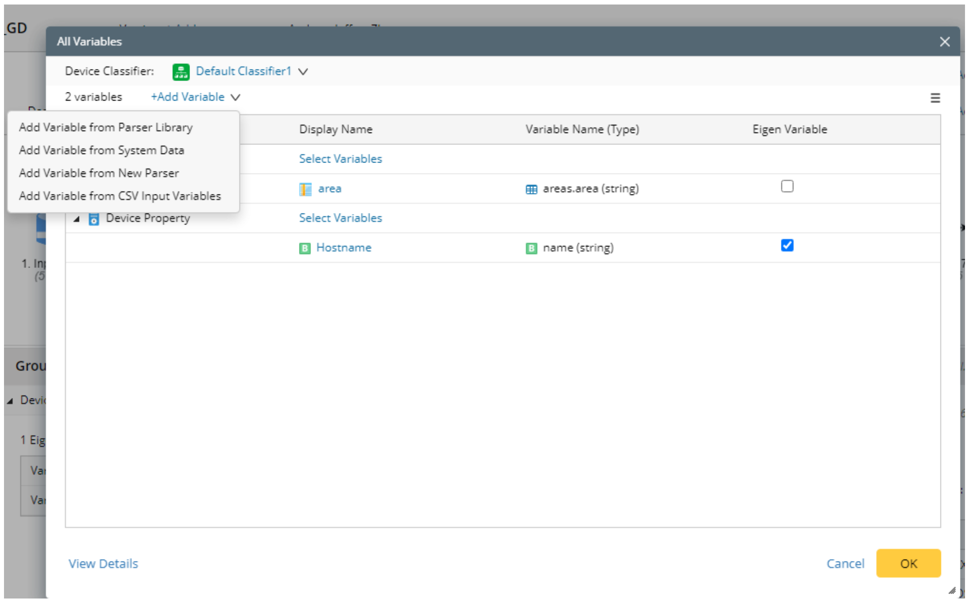
(Last executed at 2/9/2022, 2:44:36 PM) Succeeded

5 eigen groups

5 items: 5 items grouped and 0 items ungrouped Filter by: All Device Classifiers

Eigen Group	Eigen Value	Device	Device Classifier
Group 1	(US-LAX-SW2)	US-LAX-SW2	Default Classifier1
Group 5	(US-BOS-SW5)	US-BOS-SW5	Default Classifier1
Group 4	(US-BOS-SW4)	US-BOS-SW4	Default Classifier1
Group 3	(US-BOS-SW3)	US-BOS-SW3	Default Classifier1
Group 2	(US-BOS-SW2)	US-BOS-SW2	Default Classifier1

Users can add variables from the Parser library, built-in system data, and CSV input variables. Or they can create a new Parser. Under the system data, users can select the device property, interface property, and topology data.


All Variables
+Add Variable
Add Variable from Parser Library
Add Variable from System Data
Add Variable from New Parser
Add Variable from CSV Input Variables
View Details
Cancel
OK

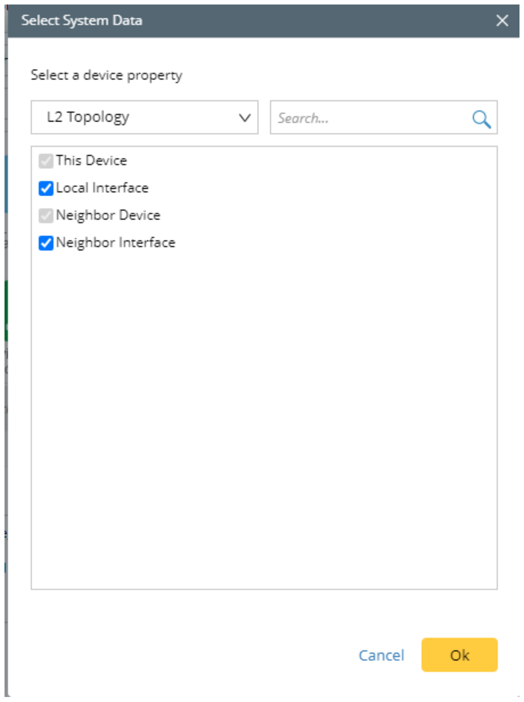
Device Classifier: Default Classifier1

2 variables

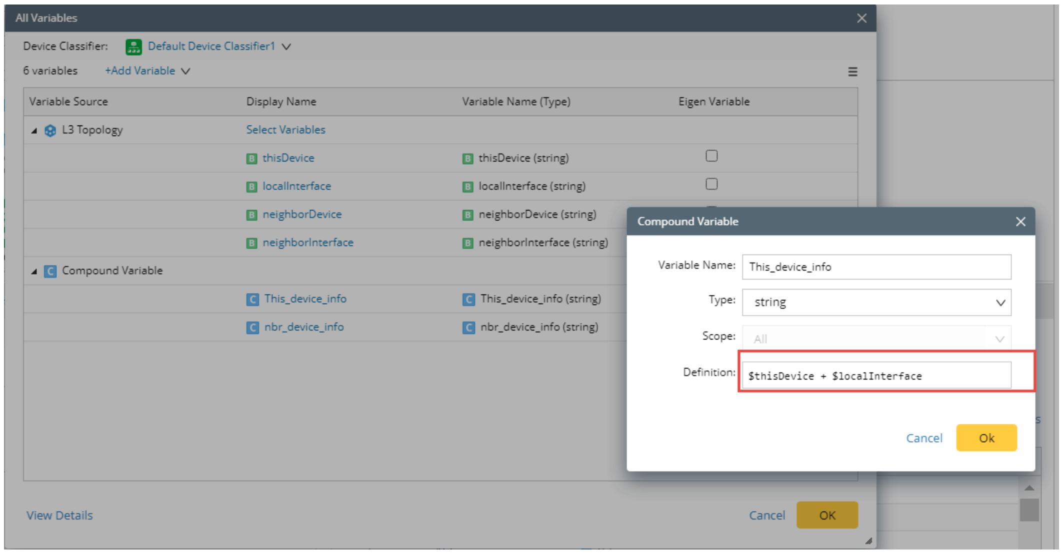
	Display Name	Variable Name (Type)	Eigen Variable
	Select Variables		
	area	areas.area (string)	☐
Device Property	Select Variables		
	Hostname	name (string)	☑

- **Compound Variables and Ignore the variable order**

While expanding a NI to check MTU mismatch between two neighbor interfaces to the whole network, users can select the topology data under the system data as the eigen variables, including four variables***, this device, local interface, neighbor device***, and ***neighbor interface***.



Furthermore, users can add **compound variables** built from the currently selected variables. For example, users can create a compound variable ***this_device_info*** with the formula ***$thisDevice + $localInteface.*** This compound variable will uniquely identify a local interface across the network if the device hostname is unique. Similarly, users can create the compound variable ***neigbor_device_info*** and set both compound variables as the eigen variables.


All Variables

Device Classifier: Default Device Classifier1

6 variables +Add Variable

Variable Source	Display Name	Variable Name (Type)	Eigen Variable
L3 Topology	Select Variables		
	thisDevice	thisDevice (string)	☐
	localInterface	localInterface (string)	☐
	neighborDevice	neighborDevice (string)	
	neighborInterface	neighborInterface (string)	
Compound Variable			
	This_device_info	This_device_info (string)	
	nbr_device_info	nbr_device_info (string)	

Compound Variable

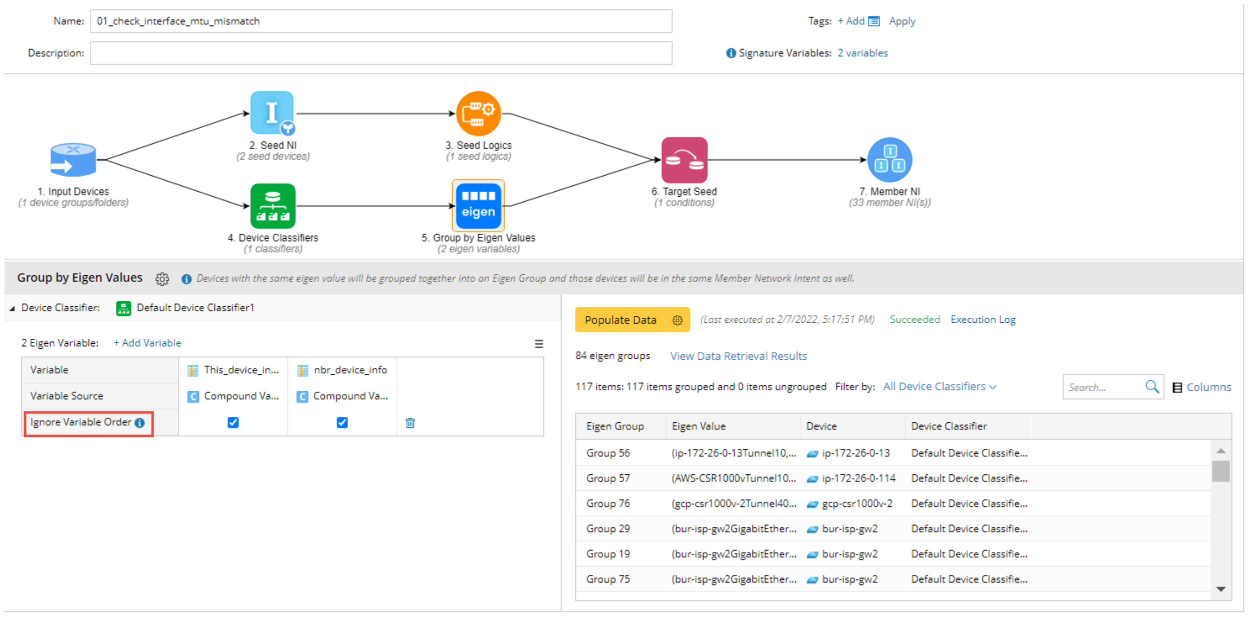
Variable Name: This_device_info

Type: string

Scope: All

Definition: `$thisDevice + $localInterface`

The system may create two eigen groups for a pair of ***this_device_info*** and ***neigbor_device_info*** as (***R1e0, R2e0***) and (***R2e0, R1e0***). However, the order is not essential for MTU mismatch, and these groups should be one. Users can ignore variable orders by adding the **Ignore Variable Order** setting and checking the corresponding variables.


Name: 01_check_interface_mtu_mismatch

Description:

Tags: + Add Apply

Signature Variables: 2 variables

Group by Eigen Values

Devices with the same eigen value will be grouped together into an Eigen Group and those devices will be in the same Member Network Intent as well.

Device Classifier: Default Device Classifier1

2 Eigen Variable: + Add Variable

Variable	This_device_in...	nbr_device_info
Variable Source	Compound Va...	Compound Va...
Ignore Variable Order	☑	☑

Populate Data *(Last executed at 2/7/2022, 5:17:51 PM)* Succeeded Execution Log

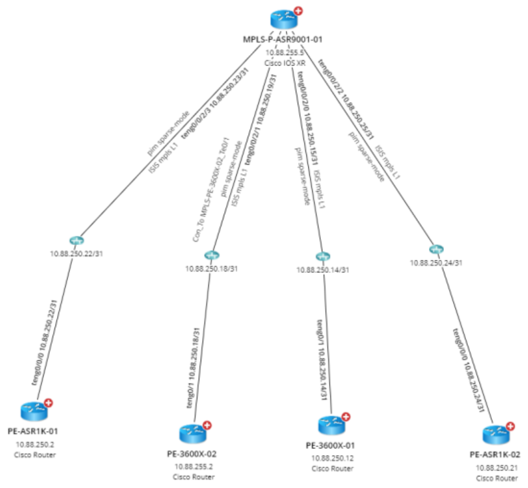
84 eigen groups View Data Retrieval Results

117 items: 117 items grouped and 0 items ungrouped Filter by: All Device Classifiers

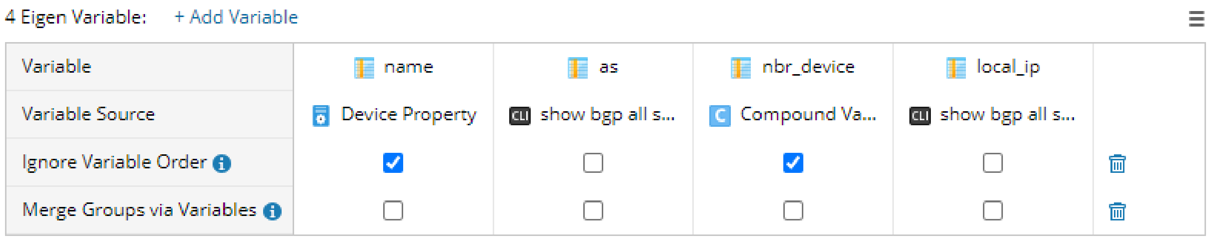
Eigen Group	Eigen Value	Device	Device Classifier
Group 56	(ip-172-26-0-13Tunnel10,...	ip-172-26-0-13	Default Device Classifie...
Group 57	(AWS-CSR1000vTunnel10...	ip-172-26-0-114	Default Device Classifie...
Group 76	(gcp-csr1000v-2Tunnel40...	gcp-csr1000v-2	Default Device Classifie...
Group 29	(bur-isp-gw2GigabitEther...	bur-isp-gw2	Default Device Classifie...
Group 19	(bur-isp-gw2GigabitEther...	bur-isp-gw2	Default Device Classifie...
Group 75	(bur-isp-gw2GigabitEther...	bur-isp-gw2	Default Device Classifie...

- **Merge group via variables**

The system will create the eigen group by default if all eigen variables are the same. In some cases, users want to group devices even if some eigen variables are not the same. For example, a NI is created to check the neighbor relationship between P and PE devices. For this purpose, we want to group the P device and its PE devices into an Eigen Group.

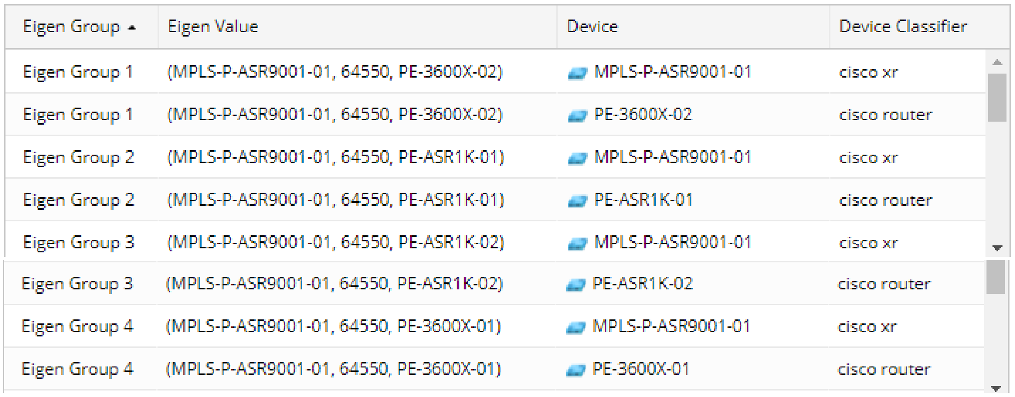


Four eigen variables are added: ***$name, $BGP_as_number, $nbr_device, $local_IP***. And the Ignore Variable Order is added to ignore the order of **name** and ***nbr_device***.



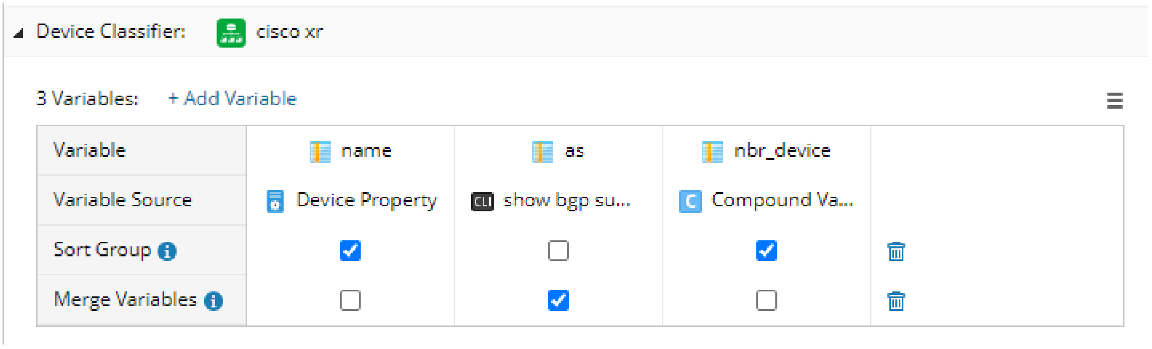
Variable	name	as	nbr_device	local_ip	
Variable Source	Device Property	show bgp all s...	Compound Va...	show bgp all s...	
Ignore Variable Order	✓	☐	✓	☐	
Merge Groups via Variables	☐	☐	☐	☐	

Four eigen groups are created for each pair of P and PE neighbors.



Eigen Group ▴	Eigen Value	Device	Device Classifier
Eigen Group 1	(MPLS-P-ASR9001-01, 64550, PE-3600X-02)	MPLS-P-ASR9001-01	cisco xr
Eigen Group 1	(MPLS-P-ASR9001-01, 64550, PE-3600X-02)	PE-3600X-02	cisco router
Eigen Group 2	(MPLS-P-ASR9001-01, 64550, PE-ASR1K-01)	MPLS-P-ASR9001-01	cisco xr
Eigen Group 2	(MPLS-P-ASR9001-01, 64550, PE-ASR1K-01)	PE-ASR1K-01	cisco router
Eigen Group 3	(MPLS-P-ASR9001-01, 64550, PE-ASR1K-02)	MPLS-P-ASR9001-01	cisco xr
Eigen Group 3	(MPLS-P-ASR9001-01, 64550, PE-ASR1K-02)	PE-ASR1K-02	cisco router
Eigen Group 4	(MPLS-P-ASR9001-01, 64550, PE-3600X-01)	MPLS-P-ASR9001-01	cisco xr
Eigen Group 4	(MPLS-P-ASR9001-01, 64550, PE-3600X-01)	PE-3600X-01	cisco router

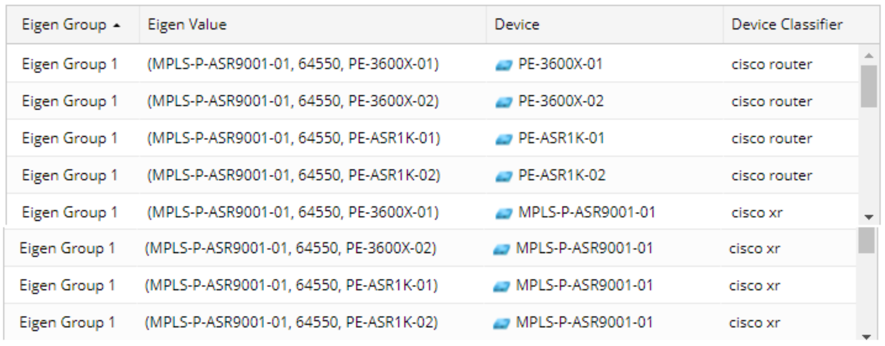
To merge all eigen groups into one group, we can enable the **merge variables** function and select the variable ***as_number*** so that the devices with the same ***as_number*** will be merged into one group.



Device Classifier: cisco xr

3 Variables: + Add Variable

Variable	name	as	nbr_device
Variable Source	Device Property	show bgp su...	Compound Va...
Sort Group	✓	☐	✓
Merge Variables	☐	✓	☐

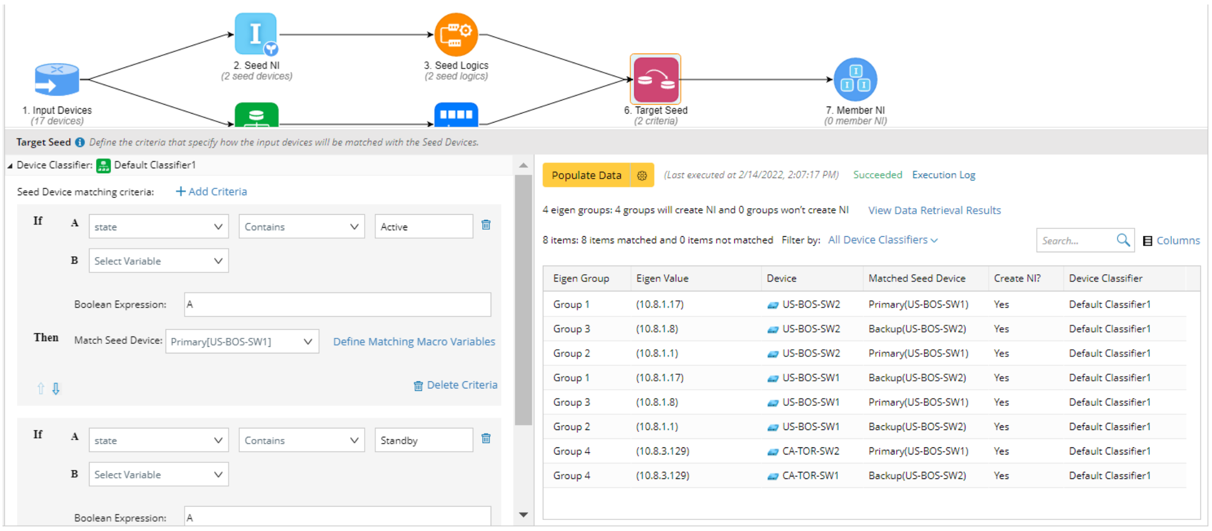


Eigen Group ▴	Eigen Value	Device	Device Classifier
Eigen Group 1	(MPLS-P-ASR9001-01, 64550, PE-3600X-01)	PE-3600X-01	cisco router
Eigen Group 1	(MPLS-P-ASR9001-01, 64550, PE-3600X-02)	PE-3600X-02	cisco router
Eigen Group 1	(MPLS-P-ASR9001-01, 64550, PE-ASR1K-01)	PE-ASR1K-01	cisco router
Eigen Group 1	(MPLS-P-ASR9001-01, 64550, PE-ASR1K-02)	PE-ASR1K-02	cisco router
Eigen Group 1	(MPLS-P-ASR9001-01, 64550, PE-3600X-01)	MPLS-P-ASR9001-01	cisco xr
Eigen Group 1	(MPLS-P-ASR9001-01, 64550, PE-3600X-02)	MPLS-P-ASR9001-01	cisco xr
Eigen Group 1	(MPLS-P-ASR9001-01, 64550, PE-ASR1K-01)	MPLS-P-ASR9001-01	cisco xr
Eigen Group 1	(MPLS-P-ASR9001-01, 64550, PE-ASR1K-02)	MPLS-P-ASR9001-01	cisco xr

2.1.1.6 Target Seed

Target Seed node defines how to match the input devices to a seed device. The system uses the **target seed** to match each device inside an **eigen group** to a **seed device** inside a **seed NI** and then from the matched seed device to match the corresponding **seed logic**. One input device can match multiple seed devices and so multiple device logic, meaning that this device can be cloned the multiple logic.

For example, an NI is created to check the failover status of a primary and backup HRSP device. The seed devices are the primary and backup devices. We can define the target seed logic: if ***$state*** contains ***Active***, match the primary seed device; if ***$state*** contains ***standby***, match the standby seed device.



Target Seed *Define the criteria that specify how the input devices will be matched with the Seed Devices.*

Device Classifier: Default Classifier1

Seed Device matching criteria: + Add Criteria

If A: state | Contains | Active
B: Select Variable
Boolean Expression: A
Then Match Seed Device: Primary[US-BOS-SW1] Define Matching Macro Variables
Delete Criteria

If A: state | Contains | Standby
B: Select Variable
Boolean Expression: A

(Last executed at 2/14/2022, 2:07:17 PM) Succeeded Execution Log

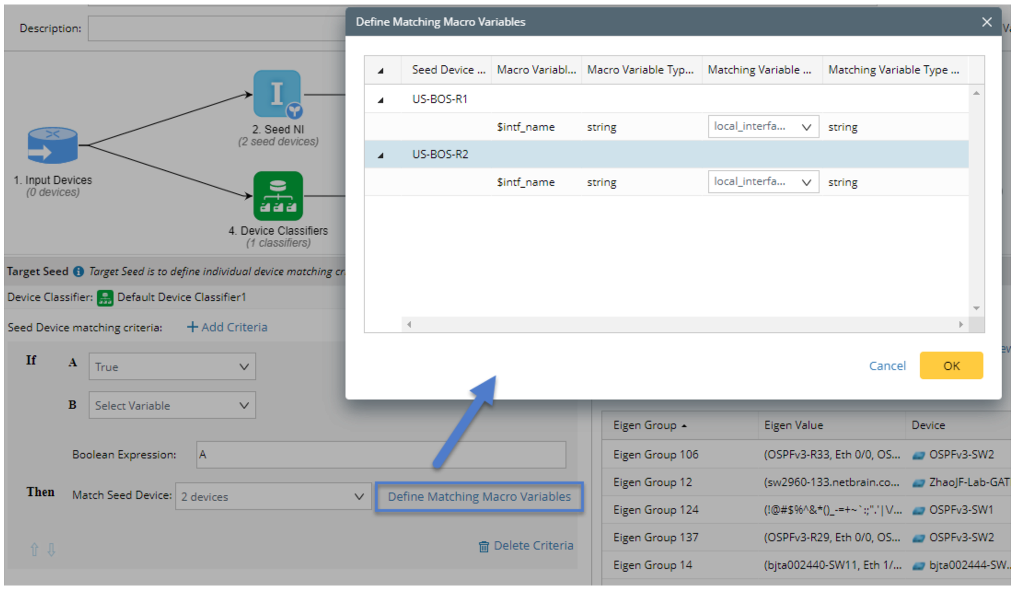
4 eigen groups: 4 groups will create NI and 0 groups won't create NI

8 items: 8 items matched and 0 items not matched Filter by: All Device Classifiers

Eigen Group	Eigen Value	Device	Matched Seed Device	Create NI?	Device Classifier
Group 1	(10.8.1.17)	US-BOS-SW2	Primary(US-BOS-SW1)	Yes	Default Classifier1
Group 3	(10.8.1.8)	US-BOS-SW2	Backup(US-BOS-SW2)	Yes	Default Classifier1
Group 2	(10.8.1.1)	US-BOS-SW2	Primary(US-BOS-SW1)	Yes	Default Classifier1
Group 1	(10.8.1.17)	US-BOS-SW1	Backup(US-BOS-SW2)	Yes	Default Classifier1
Group 3	(10.8.1.8)	US-BOS-SW1	Primary(US-BOS-SW1)	Yes	Default Classifier1
Group 2	(10.8.1.1)	US-BOS-SW1	Backup(US-BOS-SW2)	Yes	Default Classifier1
Group 4	(10.8.3.129)	CA-TOR-SW2	Primary(US-BOS-SW1)	Yes	Default Classifier1
Group 4	(10.8.3.129)	CA-TOR-SW1	Backup(US-BOS-SW2)	Yes	Default Classifier1

- **Define Matching Macro Variables**

Suppose the Seed NI has the macro variables. For example, it uses the CLI command, ***show interface e0/0*** to retrieve the data for a special interface. Users need to define which eigen variables will replace the Macro variables.



Define Matching Macro Variables

Seed Device ...	Macro Variabl...	Macro Variable Typ...	Matching Variable ...	Matching Variable Type ...
US-BOS-R1				
	$intf_name	string	local_interfa...	string
US-BOS-R2				
	$intf_name	string	local_interfa...	string

Cancel OK

Target Seed *Target Seed is to define individual device matching cr*

Device Classifier: Default Device Classifier1

Seed Device matching criteria: + Add Criteria

If A True

B Select Variable

Boolean Expression: A

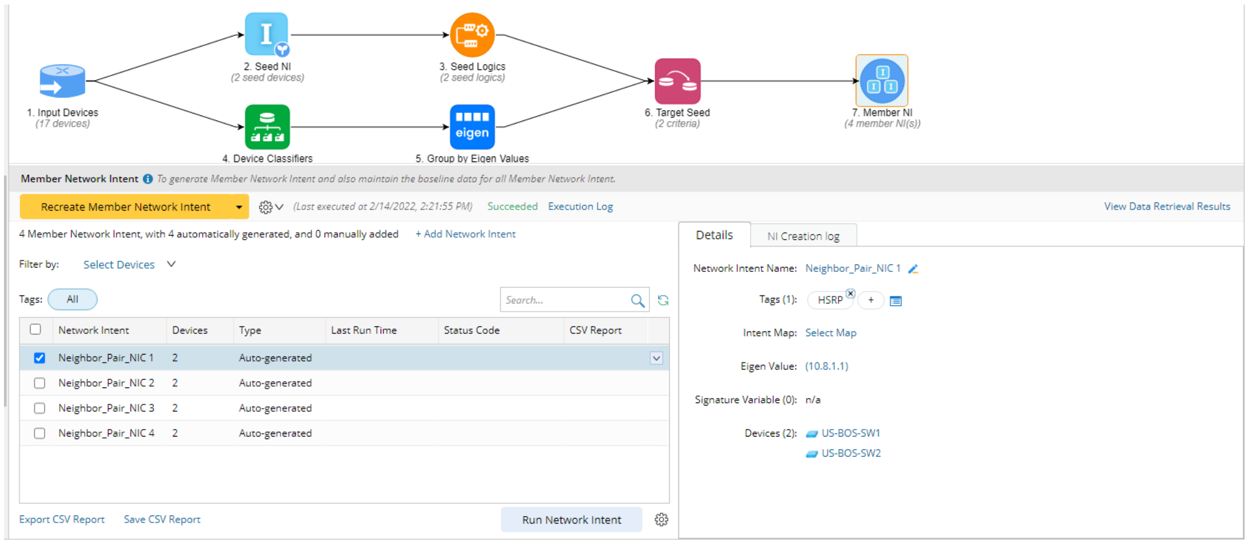
Then Match Seed Device: 2 devices Define Matching Macro Variables

Delete Criteria

Eigen Group	Eigen Value	Device
Eigen Group 106	(OSPFv3-R33, Eth 0/0, OS...	OSPFv3-SW2
Eigen Group 12	(sw2960-133.netbrain.co...	ZhaoJF-Lab-GAT
Eigen Group 124	(!@#$%^&*()_-=+~`;;".'|V...	OSPFv3-SW1
Eigen Group 137	(OSPFv3-R29, Eth 0/0, OS...	OSPFv3-SW2
Eigen Group 14	(bjta002440-SW11, Eth 1/...	bjta002444-SW...

2.1.1.7 Member NI

Member NI generates the member NIs with the following additional functions:



Member Network Intent *To generate Member Network Intent and also maintain the baseline data for all Member Network Intent.*

Recreate Member Network Intent (Last executed at 2/14/2022, 2:21:55 PM) Succeeded Execution Log View Data Retrieval Results

4 Member Network Intent, with 4 automatically generated, and 0 manually added + Add Network Intent

Filter by: Select Devices

Tags: All

Network Intent	Devices	Type	Last Run Time	Status Code	CSV Report
☑ Neighbor_Pair_NIC 1	2	Auto-generated			
☐ Neighbor_Pair_NIC 2	2	Auto-generated			
☐ Neighbor_Pair_NIC 3	2	Auto-generated			
☐ Neighbor_Pair_NIC 4	2	Auto-generated			

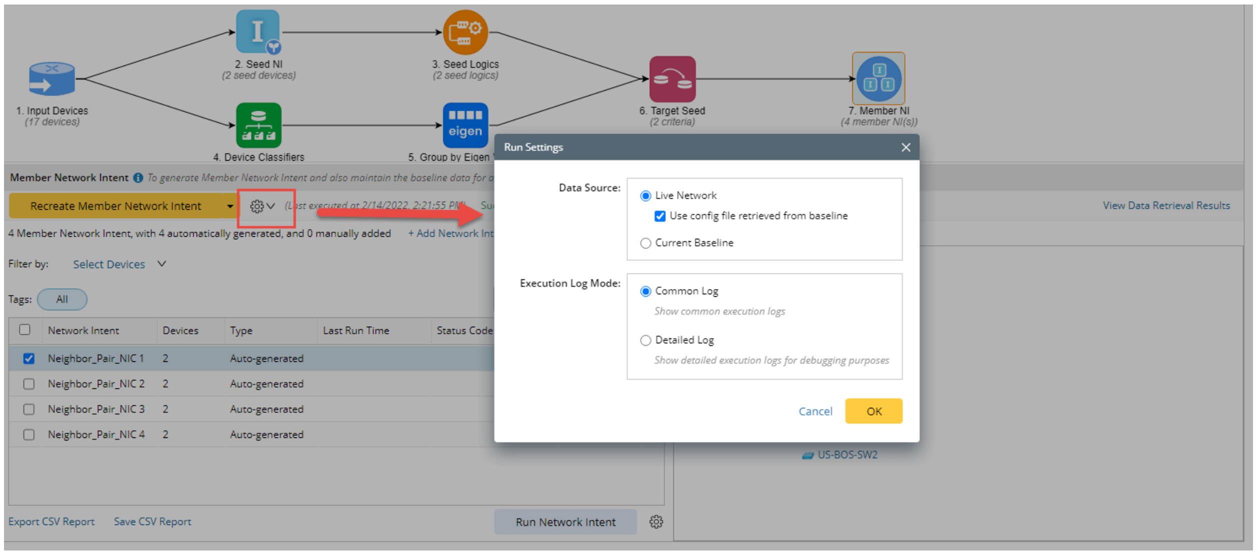
Export CSV Report Save CSV Report Run Network Intent

Details NI Creation log

Network Intent Name:	Neighbor_Pair_NIC 1
Tags (1):	HSRP +
Intent Map:	Select Map
Eigen Value:	(10.8.1.1)
Signature Variable (0):	n/a
Devices (2):	US-BOS-SW1 US-BOS-SW2

- For each member NI, users can view its member devices and eigen variables, set the Intent map, add tags, set the signature variables.

- Add the static NI as its member NIs.
- Define the run setting and setting how to create the Intent Map automatically.



Member Network Intent *To generate Member Network Intent and also maintain the baseline data for*

(Last executed at 2/14/2022, 2:21:55 PM)

4 Member Network Intent, with 4 automatically generated, and 0 manually added

Filter by: Select Devices

Tags: All

	Network Intent	Devices	Type	Last Run Time	Status Code
✓	Neighbor_Pair_NIC 1	2	Auto-generated		
	Neighbor_Pair_NIC 2	2	Auto-generated		
	Neighbor_Pair_NIC 3	2	Auto-generated		
	Neighbor_Pair_NIC 4	2	Auto-generated		

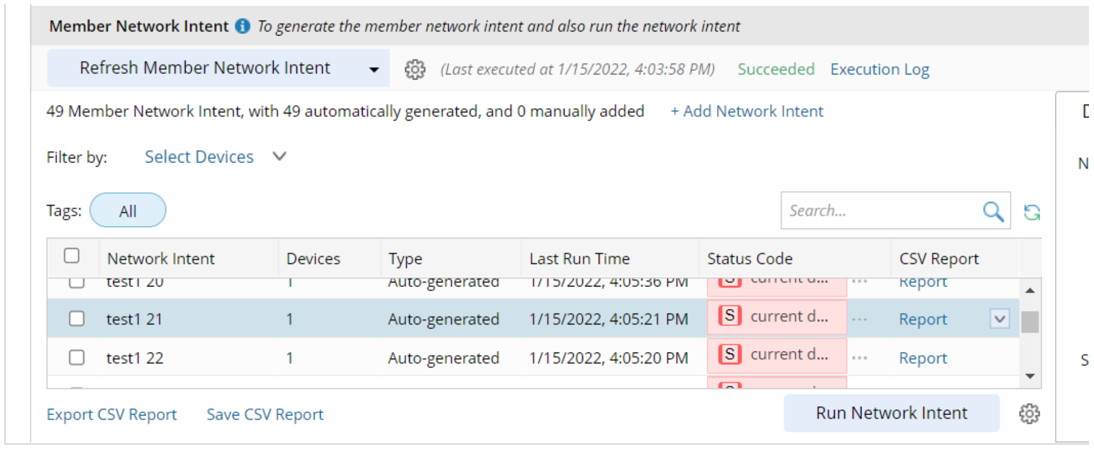
Run Settings

Data Source: [✓ Live Network] Current Baseline
[✓ Use config file retrieved from baseline]

Execution Log Mode: [✓ Common Log] Detailed Log
Show common execution logs
Show detailed execution logs for debugging purposes

Cancel OK

- Export CSV report. After executing Member NIs of a NIC, the system will merge all reports generated by member NIs and create a single report.



Member Network Intent *To generate the member network intent and also run the network intent*

(Last executed at 1/15/2022, 4:03:58 PM) Succeeded Execution Log

49 Member Network Intent, with 49 automatically generated, and 0 manually added

Filter by: Select Devices

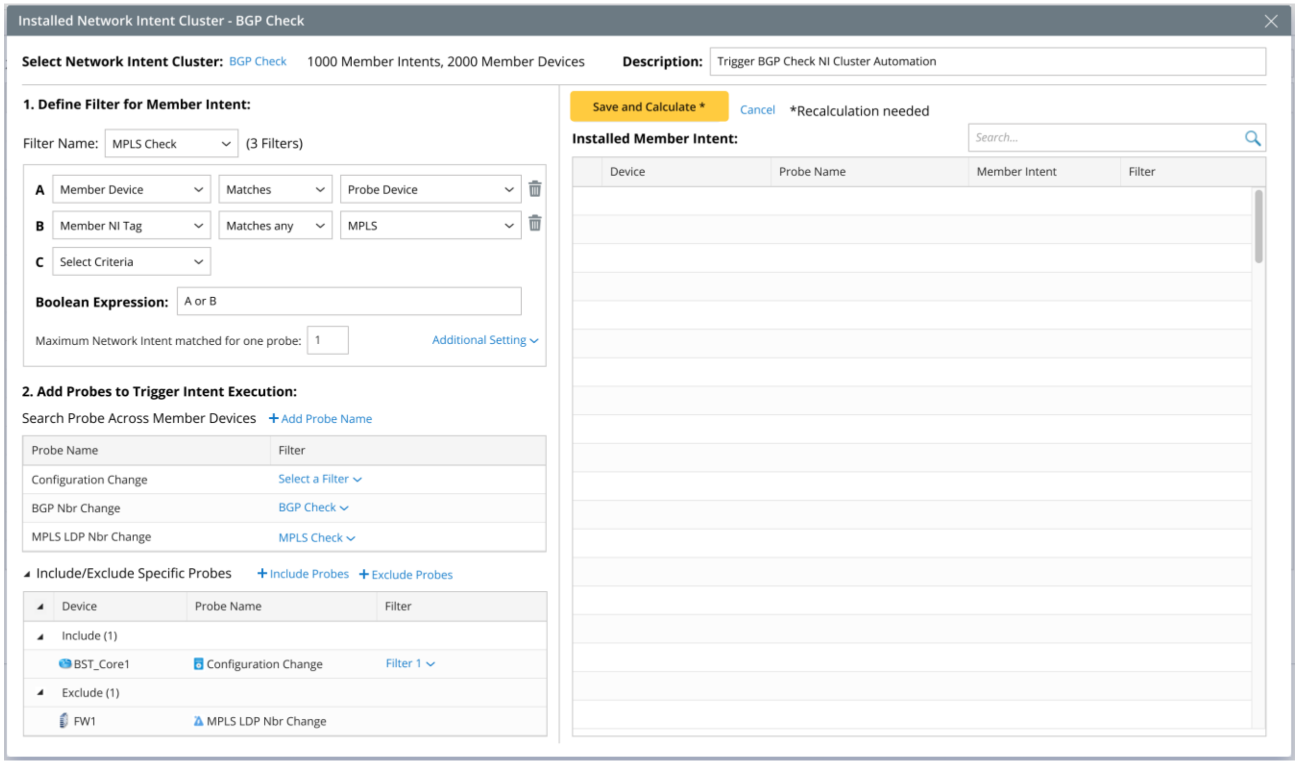
Tags: All

	Network Intent	Devices	Type	Last Run Time	Status Code		CSV Report
	test1 20	1	Auto-generated	1/15/2022, 4:05:36 PM	S current d...	...	Report
	test1 21	1	Auto-generated	1/15/2022, 4:05:21 PM	S current d...	...	Report
	test1 22	1	Auto-generated	1/15/2022, 4:05:20 PM	S current d...	...	Report

2.1.2 Execute NIC

Member NIs of a NIC can be run manually. However, the better use case is that NIC is triggered by an external ticket, which requires adding NIC to the triggered diagnosis of the TAF system, or internal probe, which requires installation of NIC to the probe.

Install NIC to a probe via three steps: 1) Select NIC; 2) Define Filter for Member Intent Member Device with Member NI Tags and signature variables; 3) Add Probe to Trigger Intent Execution.



Installed Network Intent Cluster - BGP Check

Select Network Intent Cluster: BGP Check 1000 Member Intents, 2000 Member Devices **Description:** Trigger BGP Check NI Cluster Automation

1. Define Filter for Member Intent:

Filter Name: MPLS Check (3 Filters)

A	Member Device	Matches	Probe Device
B	Member NI Tag	Matches any	MPLS
C	Select Criteria		

Boolean Expression: A or B

Maximum Network Intent matched for one probe: 1 Additional Setting

2. Add Probes to Trigger Intent Execution:

Search Probe Across Member Devices +Add Probe Name

Probe Name	Filter
Configuration Change	Select a Filter
BGP Nbr Change	BGP Check
MPLS LDP Nbr Change	MPLS Check

Include/Exclude Specific Probes +Include Probes +Exclude Probes

Device	Probe Name	Filter
Include (1)		
BST_Core1	Configuration Change	Filter 1
Exclude (1)		
FW1	MPLS LDP Nbr Change	

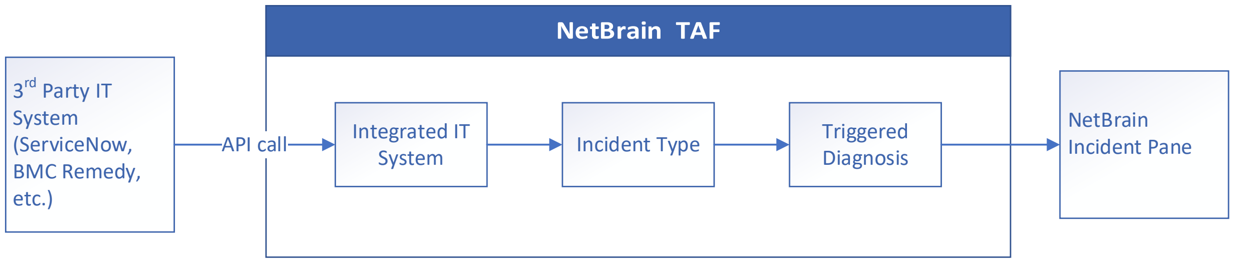
Save and Calculate * Cancel *Recalculation needed

Installed Member Intent: Search...

Device	Probe Name	Member Intent	Filter

2.2 Triggered Automation Framework

V10.1 developed a new Triggered Automation Framework (TAF), a new version of the ServiceNow App, and enhanced the Incident Pane to better support PDAs.



TAF has the following key components:

1. Integrated IT System: define the categories of API calls and what data for each API call comes from the IT system (ticket system) to be integrated with NetBrain.
2. Incident Type: for each category of the incoming API call from the Integrated IT Systems, TAF will further classify them into NetBrain Incident types. The Incident Type defines:
    a) The condition to put an API call into this Incident Type.
    b) The signature to decide whether merge the API call into an existing Incident or create a new Incident.
    c) The Incident message and Guidebook, which will be displayed in the Incident Pane.
3. Triggered Diagnosis: each Incident Type can be installed to execute NI/NIC. The installed NI and NIC can be run automatically (triggered Diagnosis) by the incoming API call or displayed in Incident Pane for the user to execute manually (self-service). The Diagnosis results and NI codes are shown in Incident Pane and the Integrated IT system. The Triggered Diagnosis defines:
    a) When to trigger run NI/NIC (triggered condition).
    b) Which member NIs for a NIC (member Network Intent filter).
    c) How to run a member NI (member NI execution mode). A user can select create the Intent Map only, Execute the NI only, or both.

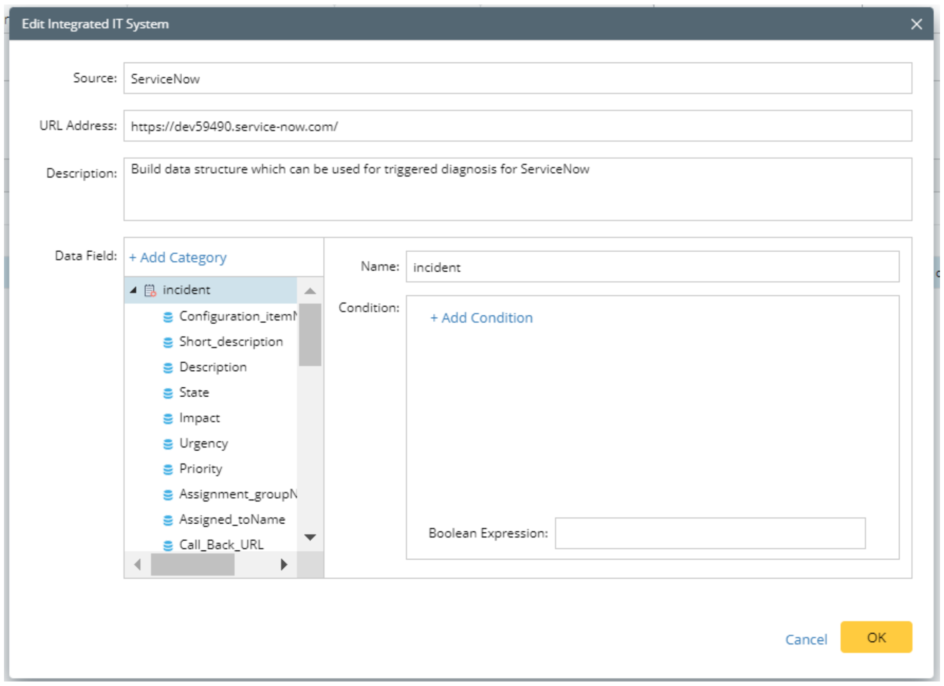
2.2.1 Integrated IT System

The first step of integrating an IT system is to define the API call signature (or identification) from that system to the NetBrain IE system. This can be done via defining an Integrated IT System at the system management level. An Integrated IT system describes what types of API calls (**category**) and the data included in these API calls. In addition, the system provides a mechanism to support multi-tenant and domain deployment for MSP and other customers.

2.2.1.1 Define an Integrated IT System

An Integrated IT system has the following fields:

- **Source**: the name of the ticket system, such as ***ServiceNow***.
- **URL Address**: the URL of the ticket system, such as ***netbrain.servicenow.com***. This field is used to differentiate which source an API call is from.
- **Description**.
- **Data field**: categories of API calls and the data fields for each category.


Edit Integrated IT System

Source: ServiceNow

URL Address: https://dev59490.service-now.com/

Description: Build data structure which can be used for triggered diagnosis for ServiceNow

Data Field:

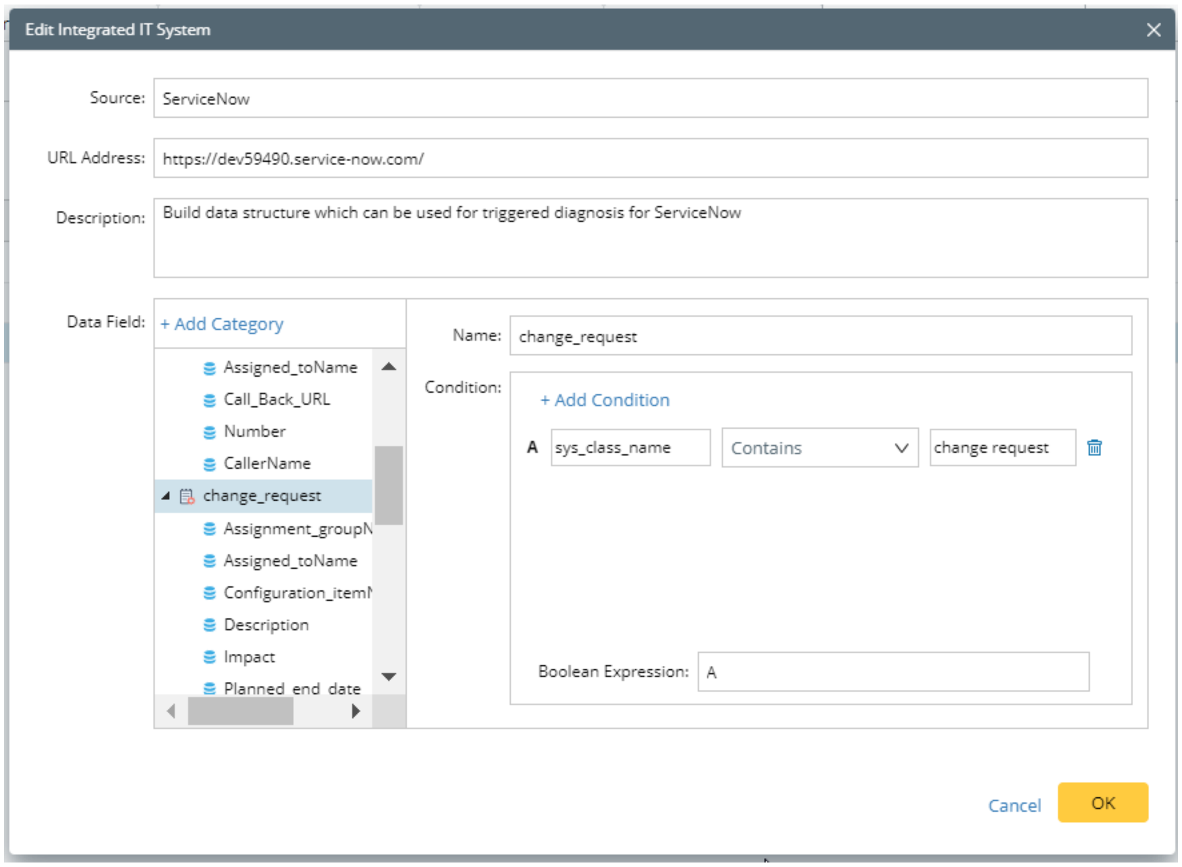
Name: incident

Condition:

Boolean Expression:

Each category corresponds to the different types of API calls from this ticket system, which usually has various data fields or parameters. For example, one category for the Incident ticket and another category for the Change Request ticket. Users can manually add data fields or import from a JSON file.

If multiple categories are defined for an IT system, TAF needs to match an API call to a category by looking for a particular data field, ***category,*** of the API call. Therefore, we recommend that the user add this particular data field to all categories. Otherwise, a user can define a condition of a category used by TAF to tell which category an incoming API call from this ticket system belongs to.


Edit Integrated IT System

Source: ServiceNow

URL Address: https://dev59490.service-now.com/

Description: Build data structure which can be used for triggered diagnosis for ServiceNow

Data Field: + Add Category

- Assigned_toName
- Call_Back_URL
- Number
- CallerName
- change_request
  - Assignment_groupN
  - Assigned_toName
  - Configuration_itemN
  - Description
  - Impact
  - Planned_end_date

Name: change_request

Condition: + Add Condition

A sys_class_name Contains change request

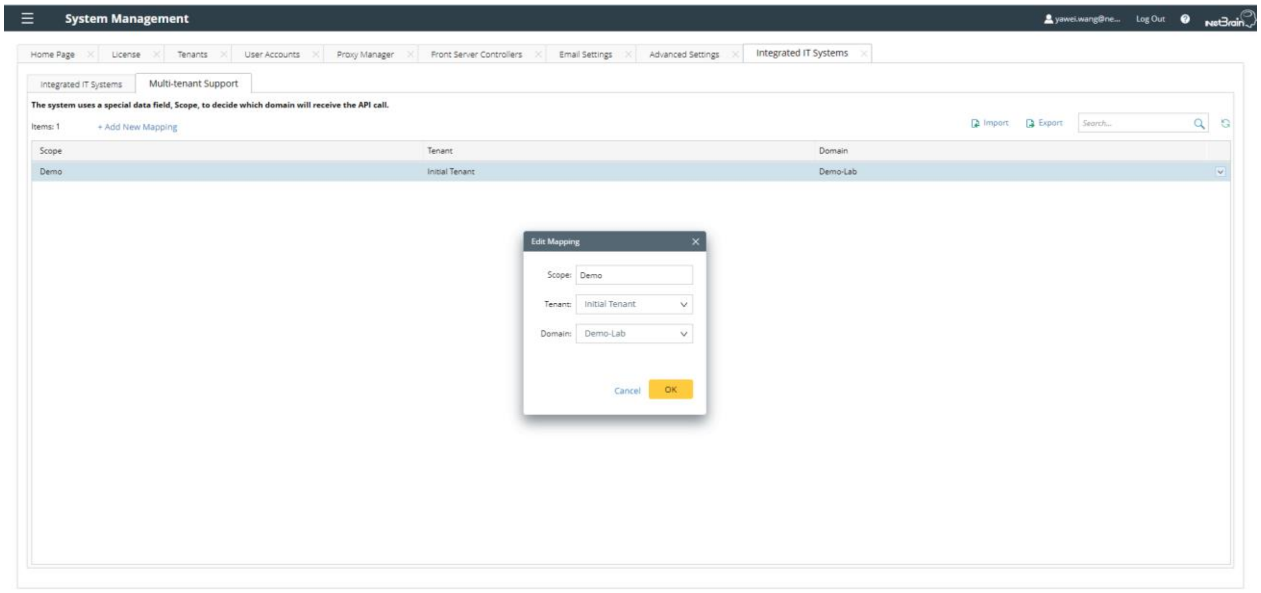
Boolean Expression: A

Cancel OK

ServiceNow App 3.0 provides the ***Send Data Fields to NetBrain*** feature, automatically creating or updating the integrated IT System for the ServiceNow. In addition, the auto-created Integration IT System includes the particular field, ***category***.

2.2.1.2 Multi-tenant Support

MSP customers usually have multiple tenant systems, one tenant for one client. To support the multi-tenant/Domain, an API call must include a particular data field, ***scope,*** and define mappings between scopes and Domains for Integrated IE systems. TAF framework will forward the API call to the matched domain.



The system uses a special data field, Scope, to decide which domain will receive the API call.

Items: 1

Scope	Tenant	Domain
Demo	Initial Tenant	Demo-Lab

Edit Mapping

Scope: Demo

Tenant: Initial Tenant

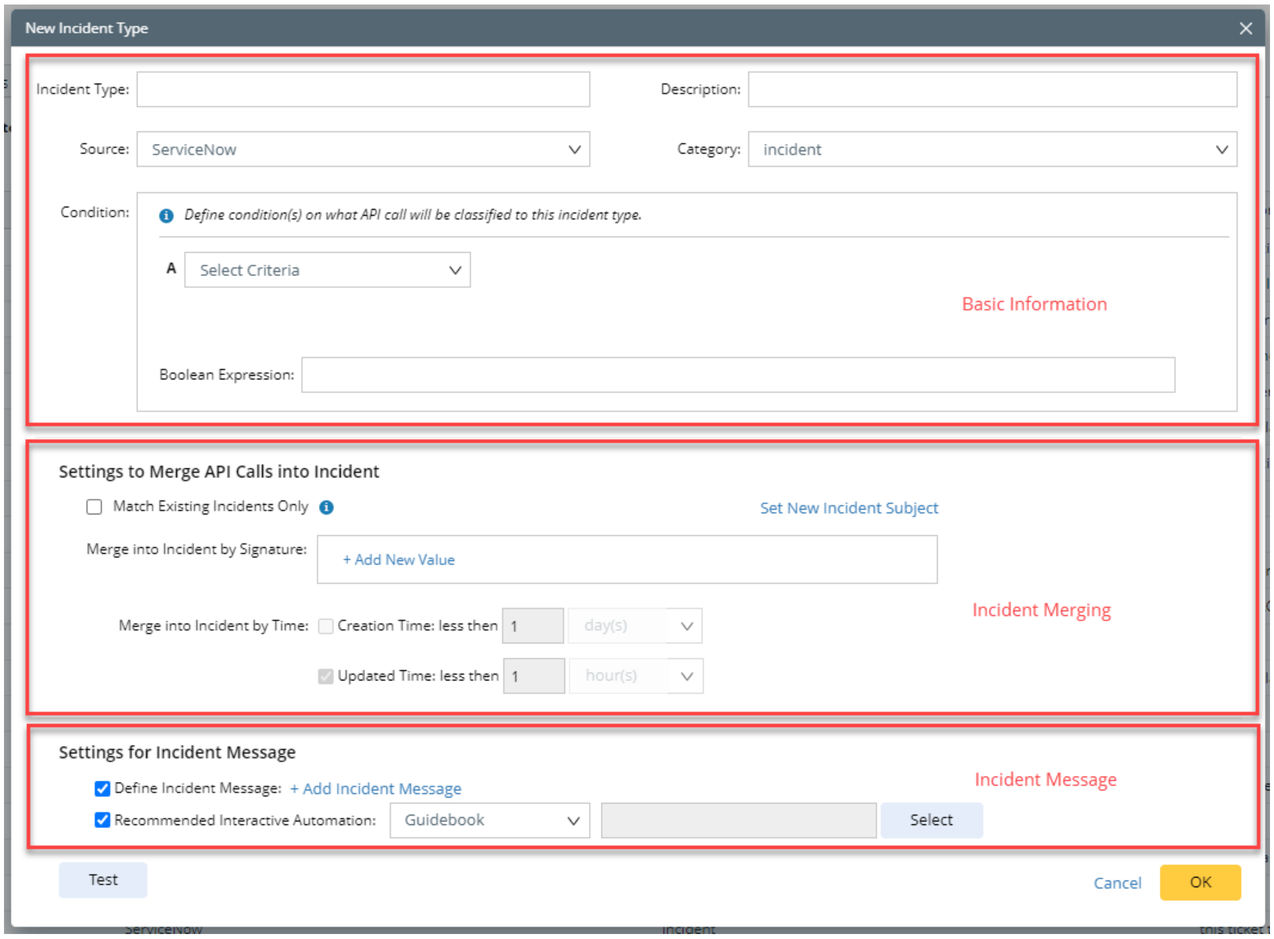
Domain: Demo-Lab

2.2.2 Define and Test Incident Type

For each category of the incoming API call from the Integrated IT Systems, TAF will further classify them into NetBrain Incident types. The Incident Type defines:

1. The condition to put an API call into this Incident Type.
2. The signature variables to decide whether merge the API call into an existing Incident or create a new Incident.
3. The Incident message and Guidebook, which will be displayed in the Incident Pane.

The definition of Incident type has three steps:


New Incident Type

Incident Type:

Description:

Source: ServiceNow

Category: incident

Condition: Define condition(s) on what API call will be classified to this incident type.

A Select Criteria

Boolean Expression:

Settings to Merge API Calls into Incident

Match Existing Incidents Only

Merge into Incident by Signature: + Add New Value

Merge into Incident by Time: Creation Time: less then 1 day(s)

Updated Time: less then 1 hour(s)

Settings for Incident Message

Define Incident Message: + Add Incident Message

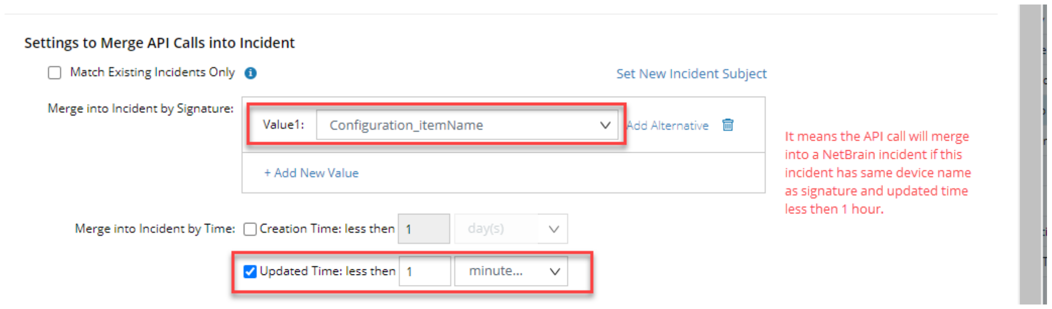
Recommended Interactive Automation: Guidebook

1. **Basic settings**

- **Incident Type**: a unique name, such as ***Interface Error, BGP Down***, etc.
- **Description**: an optional field to describe the Incident.
- **Source**: select an Integrated IT system, such as ***ServiceNow***.
- **Category**: select a source category, such as Incident (Incident ticket from ServiceNow).
- **Condition**: define which API calls of this category coming from the source belongs to this Incident Type.

2. **Incident Merging Setting**

Often multiple tickets are related and can be caused by the same root cause. For example, if a monitoring system detects an interface is down, it may create multiple tickets. TAF allows a user to merge API calls for all these tickets into one Incident instead of creating a new incident for each of these calls. The setting to merge will be defined like this: if an API call has the same signature value as a previous API call within a specific time range, do not create a new incident. Instead, append a new Incident message to the Incident created in the last call.


Settings to Merge API Calls into Incident

Match Existing Incidents Only

Set New Incident Subject

Merge into Incident by Signature: Value1: Configuration_itemName

Add Alternative

+ Add New Value

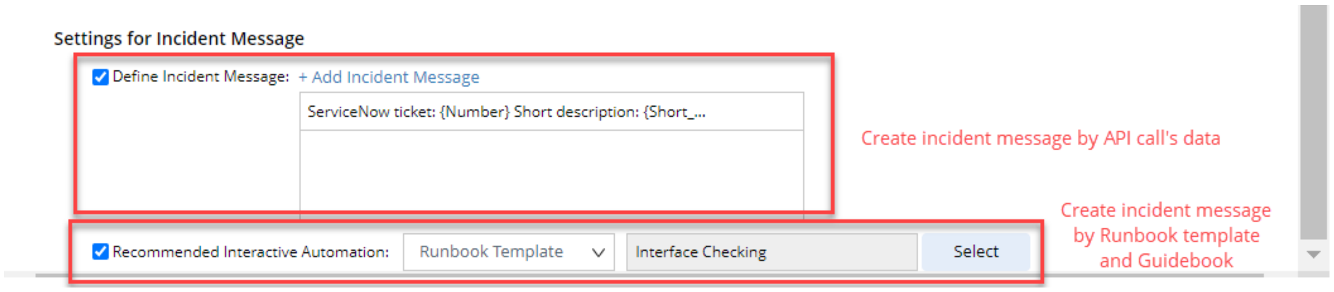
Merge into Incident by Time: Creation Time: less then 1 day(s)

✓ Updated Time: less then 1 minute...

It means the API call will merge into a NetBrain incident if this incident has same device name as signature and updated time less then 1 hour.

3. Define Incident Message

Each ticket will append a message into the corresponding Incident and optionally a recommended guidebook or Runbook template for the interactive troubleshooting. Besides the text, a user can insert any data field from the category, built-in special fields ***({Incident Type}, {source}, {category}***, and ***{triggered time}***), and hyperlink into the message.


Settings for Incident Message

✓ Define Incident Message: + Add Incident Message

ServiceNow ticket: {Number} Short description: {Short_...

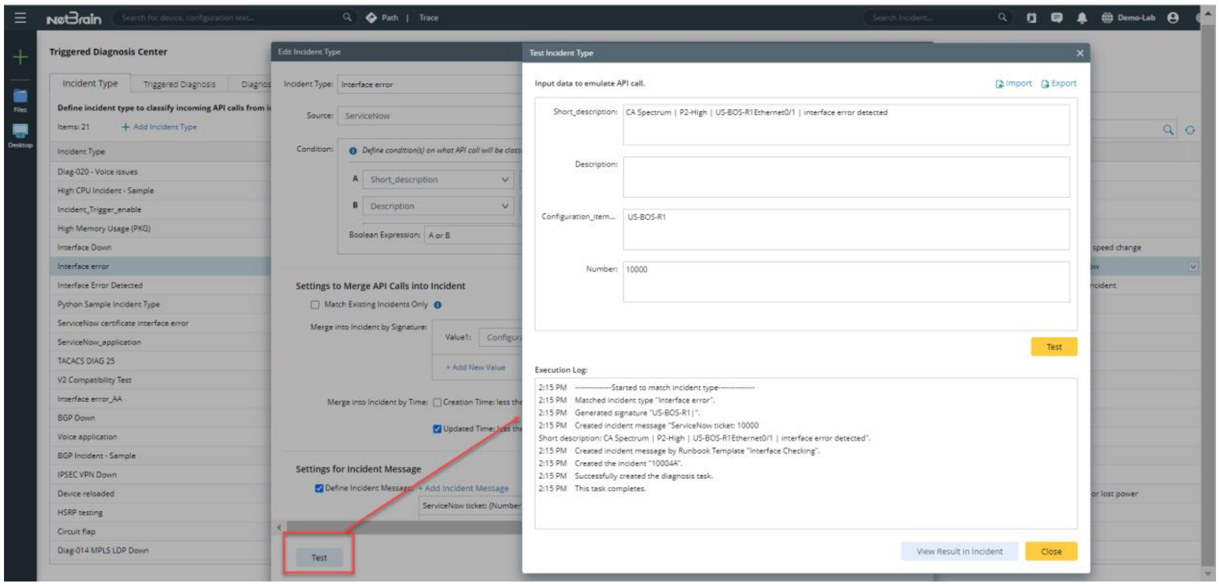
Create incident message by API call's data

✓ Recommended Interactive Automation: Runbook Template | Interface Checking | Select

Create incident message by Runbook template and Guidebook

4. Test Incident Type

Incident Type edit UI provides the **Test** button for a user to test its definition. After inputting the data fields of the incoming API call, the system prints out the execution log, including each step's details.



Test Incident Type

Input data to emulate API call.

Short_description: CA Spectrum | P2-High | US-BOS-R1Ethernet0/1 | interface error detected

Description:

Configuration_item...: US-BOS-R1

Number: 10000

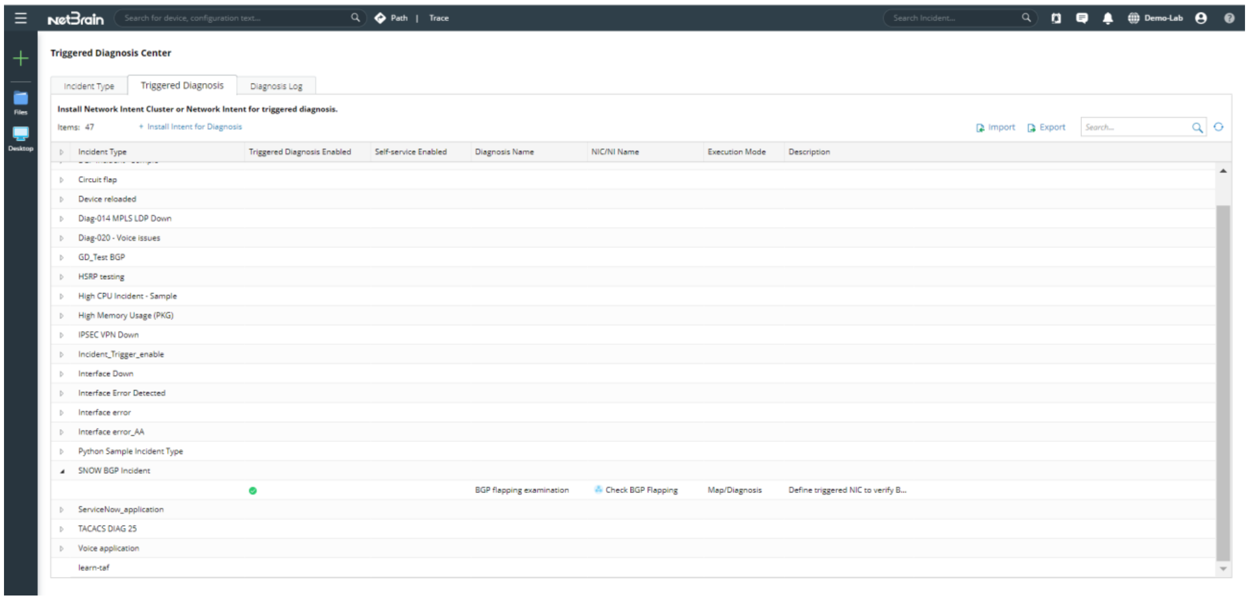
Execution Log:

```
2:15 PM   ---------------Started to match incident type---------------
2:15 PM   Matched incident type "Interface error".
2:15 PM   Generated signature "US-BOS-R1|".
2:15 PM   Created incident message "ServiceNow ticket: 10000
Short description: CA Spectrum | P2-High | US-BOS-R1Ethernet0/1 | interface error detected".
2:15 PM   Created incident message by Runbook Template "Interface Checking".
2:15 PM   Created the incident "10004A".
2:15 PM   Successfully created the diagnosis task.
2:15 PM   This task completes.
```

2.2.3 Define and Test Triggered Diagnosis

Under the ***Triggered Diagnosis*** tag of ***Triggered Diagnosis Center,*** a user can install an NI or NIC for an Incident Type. The installed NI and NIC can be run automatically (**triggered Diagnosis**) by the incoming API call or displayed in Incident Pane for the user to execute manually (**self-service**). The Diagnosis results and NI status codes are shown in Incident Pane and the Integrated IT system.

For example, you create a NIC to diagnose the BGP flapping issue, generating member NIs for all BGP devices in your network. So now, for any API call falling into BGP flapping Incident Type, you can define a Diagnosis to run this NIC when the Incident occurs. The result indicates whether a BGP flapping occurred and an Intent Map is shown to the end-user in Incident Pane or ServiceNow.



Triggered Diagnosis Center

Incident Type | Triggered Diagnosis | Diagnosis Log

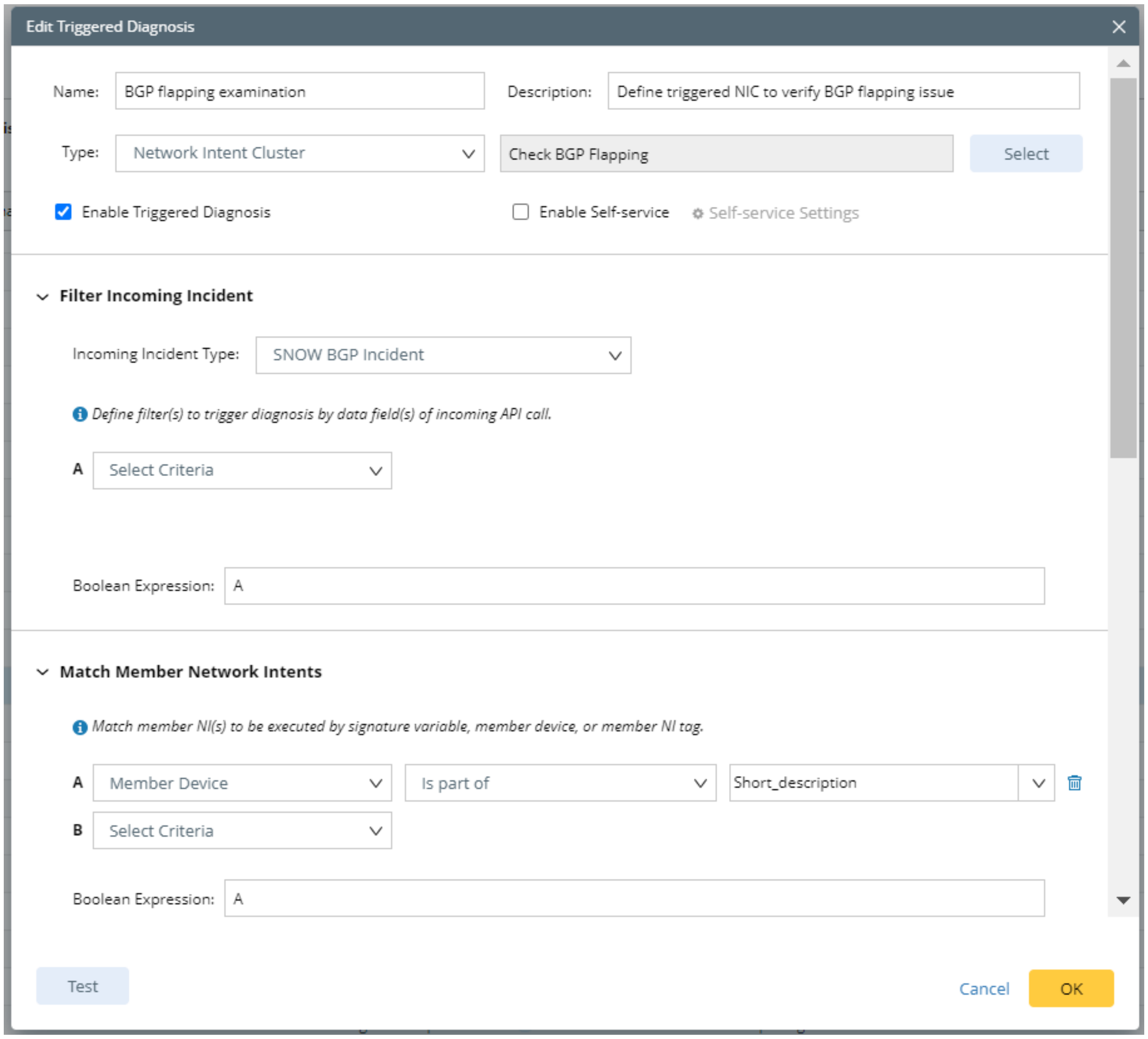
Install Network Intent Cluster or Network Intent for triggered diagnosis.

Items: 47 + Install Intent for Diagnosis

Incident Type	Triggered Diagnosis Enabled	Self-service Enabled	Diagnosis Name	NIC/NI Name	Execution Mode	Description
Circuit flap						
Device reloaded						
Diag-014 MPLS LDP Down						
Diag-020 - Voice issues						
GD_Test BGP						
HSRP testing						
High CPU Incident - Sample						
High Memory Usage (PKG)						
IPSEC VPN Down						
Incident_Trigger_enable						
Interface Down						
Interface Error Detected						
Interface error						
Interface error_AA						
Python Sample Incident Type						
SNOW BGP Incident						
	✓		BGP flapping examination	Check BGP Flapping	Map/Diagnosis	Define triggered NIC to verify B...
ServiceNow_application						
TACACS DIAG 25						
Voice application						
learn-taf						

Define a triggered diagnosis with the following steps:

1. Define the basic setting: name, description, type (NI or NIC), and select an NI/NIC
2. Enable the NI/NIC to be triggered, self-service, or both.
3. Define the conditions for the NI/NIC to be triggered (triggered condition).
4. For NIC, define which member NIs to be executed (filter member NI) and how they are executed.


Edit Triggered Diagnosis

Name: BGP flapping examination

Description: Define triggered NIC to verify BGP flapping issue

Type: Network Intent Cluster | Check BGP Flapping | Select

☑ Enable Triggered Diagnosis ☐ Enable Self-service Self-service Settings

Filter Incoming Incident

Incoming Incident Type: SNOW BGP Incident

Define filter(s) to trigger diagnosis by data field(s) of incoming API call.

A Select Criteria

Boolean Expression: A

Match Member Network Intents

Match member NI(s) to be executed by signature variable, member device, or member NI tag.

A Member Device | Is part of | Short_description

B Select Criteria

Boolean Expression: A

Test Cancel OK

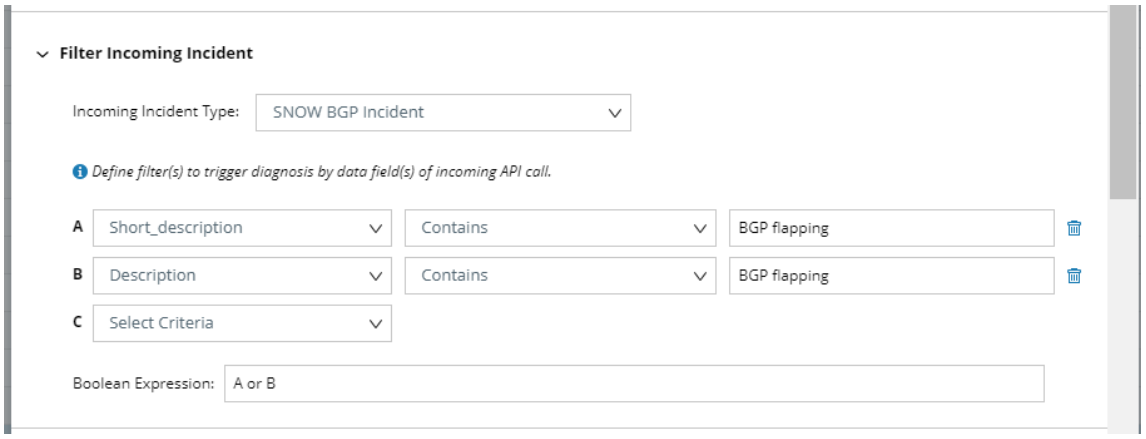
2.2.3.1 Basic Setting

Besides the name and description, you select a NI or NIC for the Diagnosis. In most cases, you should choose NIC unless the Incident Type is specific for certain devices. For example, select the NIC, ***BGP Flapping Examination***.

The NIC can be set to run automatically if the triggered condition is satisfied or displayed in the Incident Portal for the end-user can manually run it (self-service).

2.2.3.2 Triggered Condition

Trigged condition defines when this Diagnosis will be executed. First, you select the Incident Type to trigger this Diagnosis. Then optionally define the condition. If no condition is specified, the Diagnosis will always be executed when the incoming API call belongs to the Incident Type.



Filter Incoming Incident

Incoming Incident Type: SNOW BGP Incident

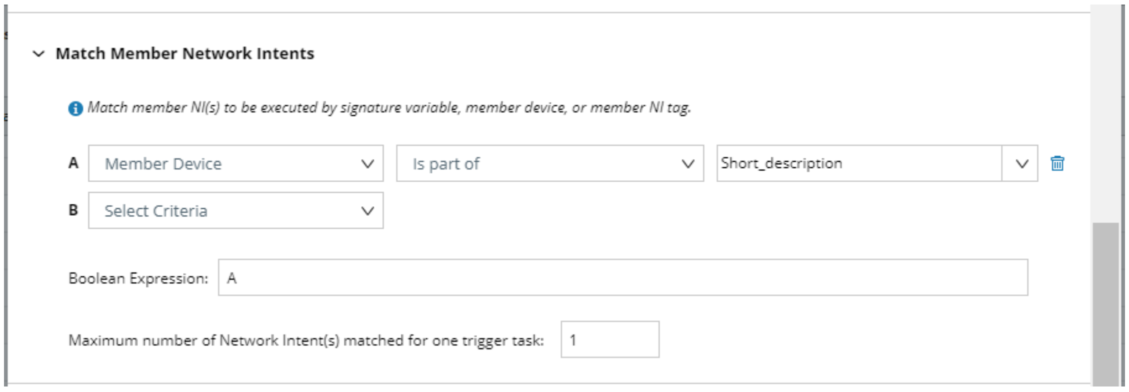
Define filter(s) to trigger diagnosis by data field(s) of incoming API call.

A	Short_description	Contains	BGP flapping
B	Description	Contains	BGP flapping
C	Select Criteria		

Boolean Expression: A or B

2.2.3.3 Filter for Member Network Intent

For NIC diagnosis, users can filter the member NIs to be executed. In our example, you may not run all member NIs if there are many BGP devices in your network. Instead, you may want to run the member NIs for the device(s) related to this Incident.



Match Member Network Intents

Match member NI(s) to be executed by signature variable, member device, or member NI tag.

A	Member Device	Is part of	Short_description
B	Select Criteria		

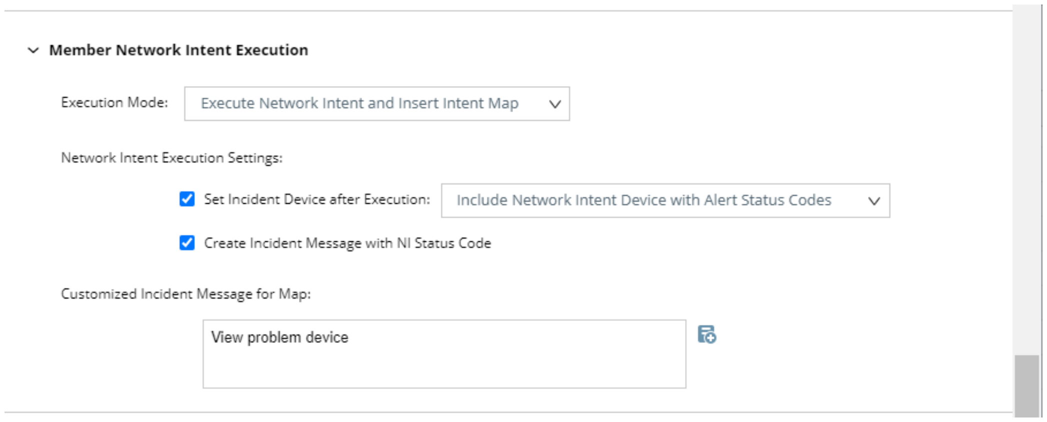
Boolean Expression: A

Maximum number of Network Intent(s) matched for one trigger task: 1

You may set Maximum Network Intent Matched for One Trigger to be a reasonable number to protect the system.

2.2.3.4 Member Network Intent Execution

NIC defines the logic to check the network state against the Intent and create a map for the Intent. The user can specify which to execute.



Member Network Intent Execution

Execution Mode: Execute Network Intent and Insert Intent Map

Network Intent Execution Settings:

☑ Set Incident Device after Execution: Include Network Intent Device with Alert Status Codes

☑ Create Incident Message with NI Status Code

Customized Incident Message for Map:

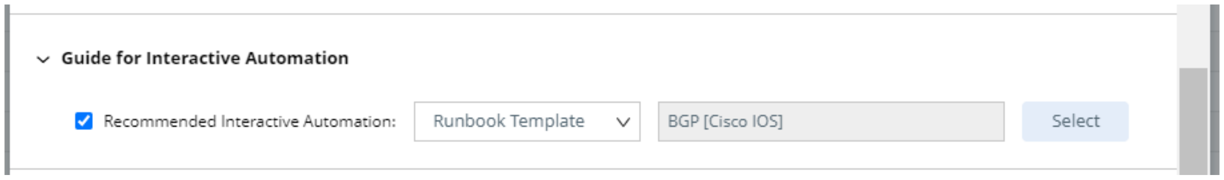
View problem device

- Select the Execution mode: **Execute Network Intent Only**, **Insert Intent Map Only**, and **Execute Network Intent and Add Intent Map**.
- Select the Network Intent Setting: defining how the results are displayed in the Incident Portal. The option ***Set Incident Device*** after Execution allows the user to set the incident device(s) to include all Network Intent devices or only the Network Intent Devices with the alert status codes. The option ***Create Incident Message by Status Code*** will create an Incident message with the status code.
- If an NI has an Intent Map, the system will display the map in Incident Portal. Otherwise, the system will create an Intent Map according to the logic defined in NIC. An incident message will also be created with a hyperlink of the Intent Map.

This setting affects both the triggered and self-service Diagnosis. If ***Execute Network Intent*** is selected, this Diagnosis will be available for the manual Trigger NetBrain Diagnosis (in the Integrated IT system and Incident Portal). Similarly, if ***Insert Intent Map*** is selected, the Diagnosis will be available for the manual Trigger NetBrain Map (in the Integrated IT system and Incident Portal).

2.2.3.5 Guide for Interactive Automation

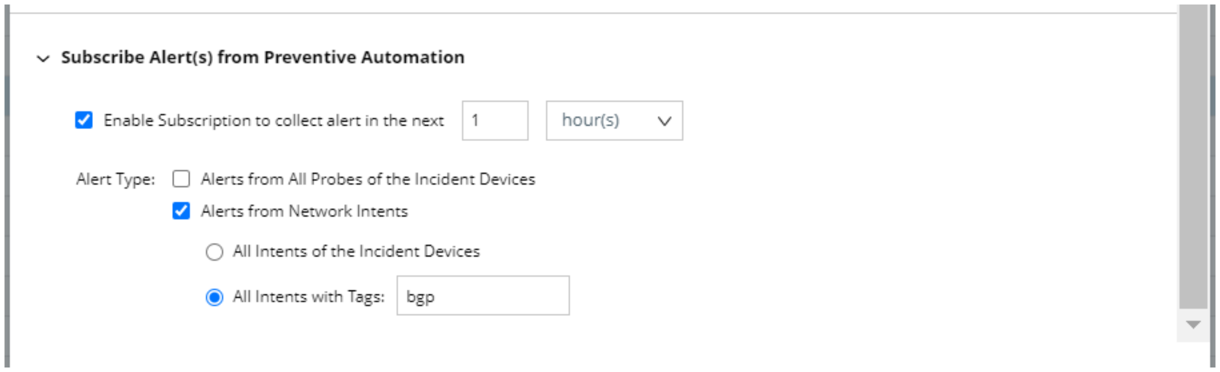
The user can select a guidebook or a Runbook Template to guide the end-user to run the recommended automation in the Incident Portal.


Guide for Interactive Automation

☑ Recommended Interactive Automation: Runbook Template | BGP [Cisco IOS] | Select

2.2.3.6 Subscribe to Preventive Automation

A diagnosis can be configured to collect the alerts from Flash Probe and/or NIs. The user can define the time range (e.g., next one day), filter tag (e.g., BGP probe or NI), and alert type from Intent.


Subscribe Alert(s) from Preventive Automation

☑ Enable Subscription to collect alert in the next 1 hour(s)

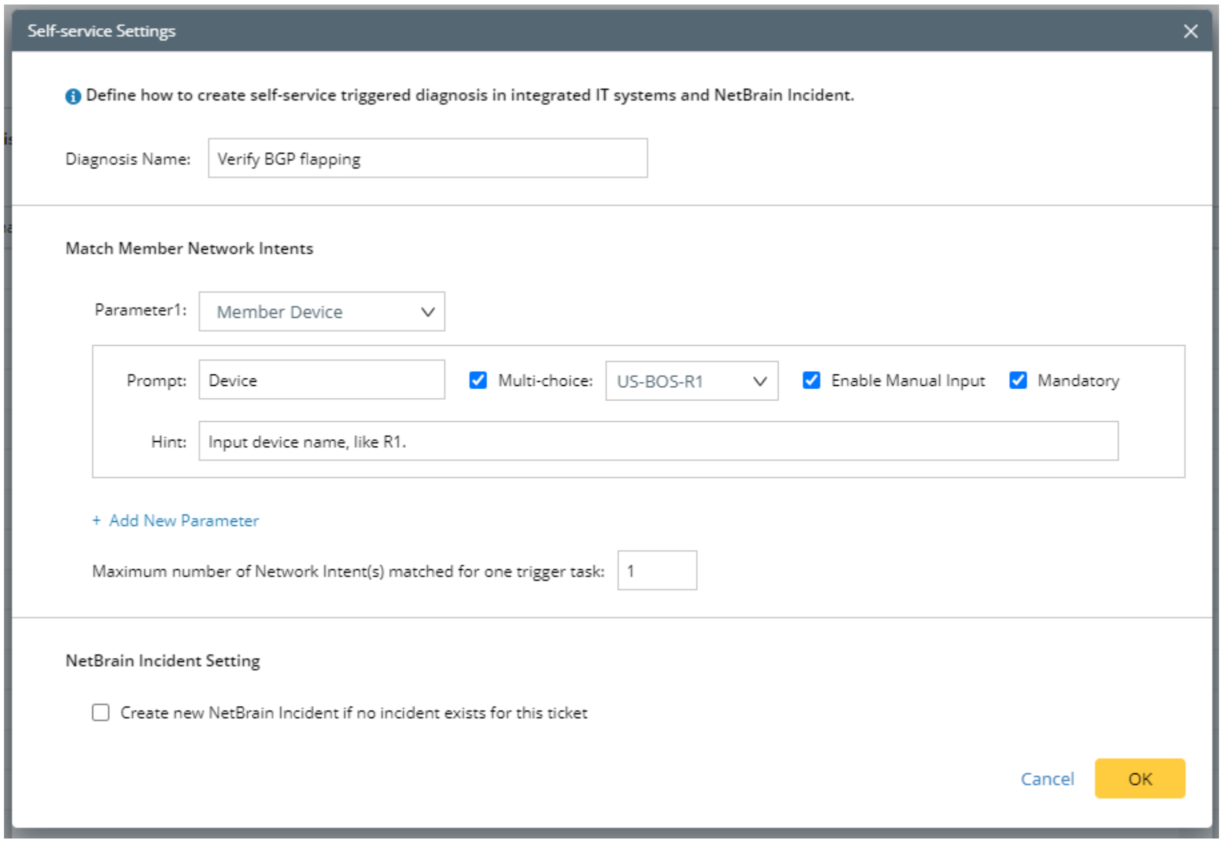
Alert Type: ☐ Alerts from All Probes of the Incident Devices
☑ Alerts from Network Intents
○ All Intents of the Incident Devices
● All Intents with Tags: bgp

The system will collect all alerts from the fresh probe or NIs on all incident devices in the configured time range and display them in Incident Pane.

2.2.3.7 Self-service Settings

If a diagnosis is enabled for Self-service, an end-user can select and run this Diagnosis manually from Incident Portal or the IT Integrated System such as ServiceNow. The Self Service enables Network engineers to share essential diagnosis functions with IT engineers. See section 2.3.5 and section 2.4.1 on how users can run the self-service diagnosis in ServiceNow and Incident Pane correspondingly.

Self-service Settings defines parameters an end-user must input in the popup window when the Diagnosis is selected and run and other options:


Self-service Settings

Define how to create self-service triggered diagnosis in integrated IT systems and NetBrain Incident.

Diagnosis Name: Verify BGP flapping

Match Member Network Intents

Parameter1: Member Device

Prompt: Device ☑ Multi-choice: US-BOS-R1 ☑ Enable Manual Input ☑ Mandatory

Hint: Input device name, like R1.

+ Add New Parameter

Maximum number of Network Intent(s) matched for one trigger task: 1

NetBrain Incident Setting

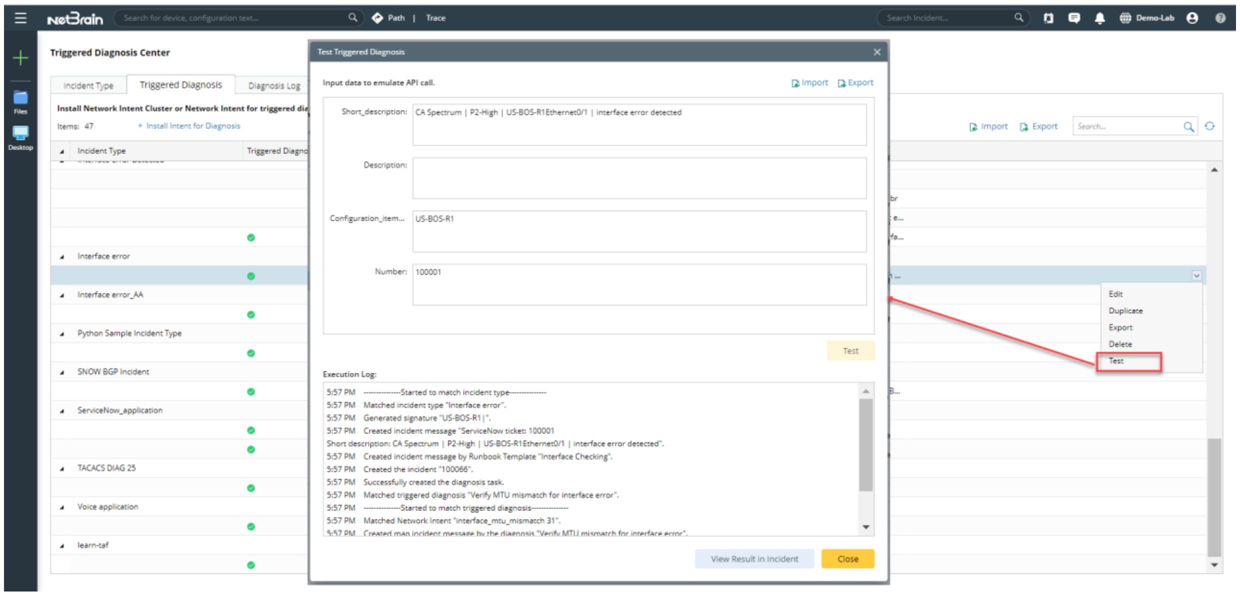
☐ Create new NetBrain Incident if no incident exists for this ticket

- **Diagnosis name**: the name displayed to the end-user while selecting a diagnosis to run. It can be different than the diagnosis name defined in the Triggered Diagnosis window.
- **Parameters to filter the member NIs**: the user can select NIC signature variable, member device, member network intent tag. For each parameter, the user can define the prompt, whether the end-user selects the value from the multiple-choice and/or enters the value manually, whether it is mandatory, and hint. When **Multi-choice** is enabled, the user should enter the possible choices separated by the semi-colon (;). These choices will be displayed to the end-user as the dropdown menu. If both multiple-choice and manual input is enabled, the end-user can manually enter the value or select from the dropdown list.
- ***Maximum Network Intent Matched for One Trigger*** defines the maximum of matched NIs. The system will stop matching NIs when this number is reached.
- Checkbox ***Create New NetBrain Incident if No Incident Exists in this ticket*** will create a new incident if no incident exists for this ticket.

The self-service setting has the default values so that the Diagnosis can typically work if a user does not change the default setting.

2.2.3.8 Test Triggered Diagnosis

Triggered Diagnosis UI provides the **Test** button for a user to test its definition. After inputting the data fields to emulate the API call, the system prints out the execution log with each step's details.



Input data to emulate API call.

Short_description:	CA Spectrum | P2-High | US-BOS-R1Ethernet0/1 | interface error detected
Description:	
Configuration_item...	US-BOS-R1
Number:	100001

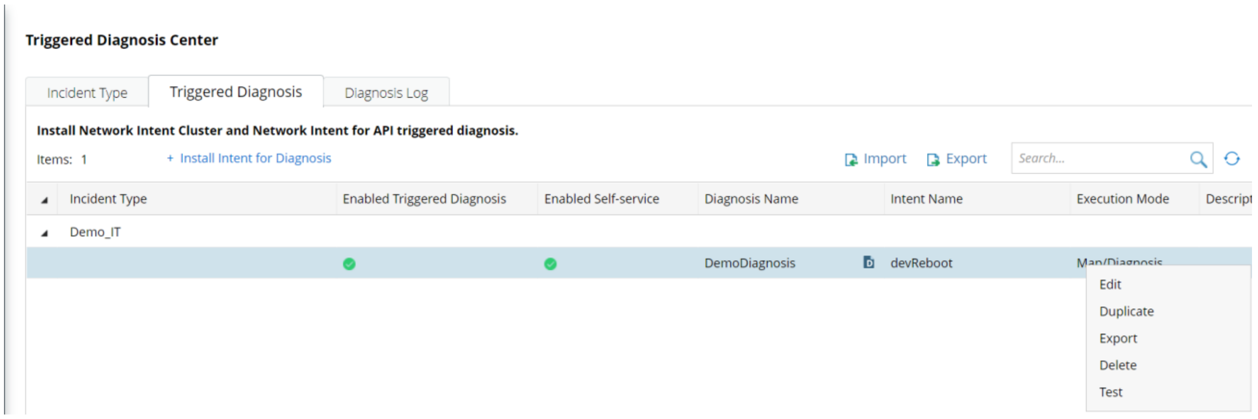
Execution Log:

```
5:57 PM   ---------------Started to match incident type---------------
5:57 PM   Matched incident type "Interface error".
5:57 PM   Generated signature "US-BOS-R1|".
5:57 PM   Created incident message "ServiceNow ticket: 100001
Short description: CA Spectrum | P2-High | US-BOS-R1Ethernet0/1 | interface error detected".
5:57 PM   Created incident message by Runbook Template "Interface Checking".
5:57 PM   Created the incident "100066".
5:57 PM   Successfully created the diagnosis task.
5:57 PM   Matched triggered diagnosis "Verify MTU mismatch for interface error".
5:57 PM   ---------------Started to match triggered diagnosis---------------
5:57 PM   Matched Network Intent "interface_mtu_mismatch 31".
5:57 PM   Created map incident message by the diagnosis "Verify MTU mismatch for interface error".
```

A link is provided to view the Incident for this incoming call.

2.2.3.9 Manage Triggered Diagnosis

Triggered Diagnosis is managed in the ***Triggered Diagnosis Center***, where a user can view all Diagnoses grouped by Incident Types, create, edit, delete, and duplicate (copy) a diagnosis. The center also provides the standard search and import/export functions.



Triggered Diagnosis Center

Incident Type | Triggered Diagnosis | Diagnosis Log

Install Network Intent Cluster and Network Intent for API triggered diagnosis.

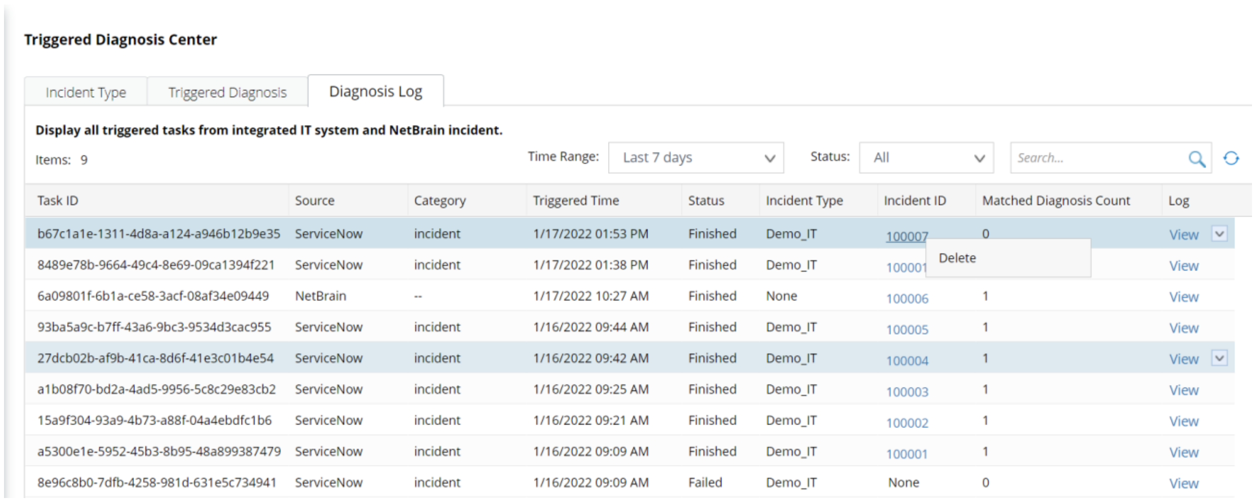
Items: 1 + Install Intent for Diagnosis

Import Export Search...

Incident Type	Enabled Triggered Diagnosis	Enabled Self-service	Diagnosis Name	Intent Name	Execution Mode	Descript
Demo_IT						
	●	●	DemoDiagnosis	devReboot	Map/Diagnosis	

2.2.4 Triggered Diagnosis Log

Under ***Triggered Diagnosis Log*** tag of ***Triggered Diagnosis Center***, logs of all triggered Diagnosis are listed,



Triggered Diagnosis Center

Incident Type | Triggered Diagnosis | Diagnosis Log

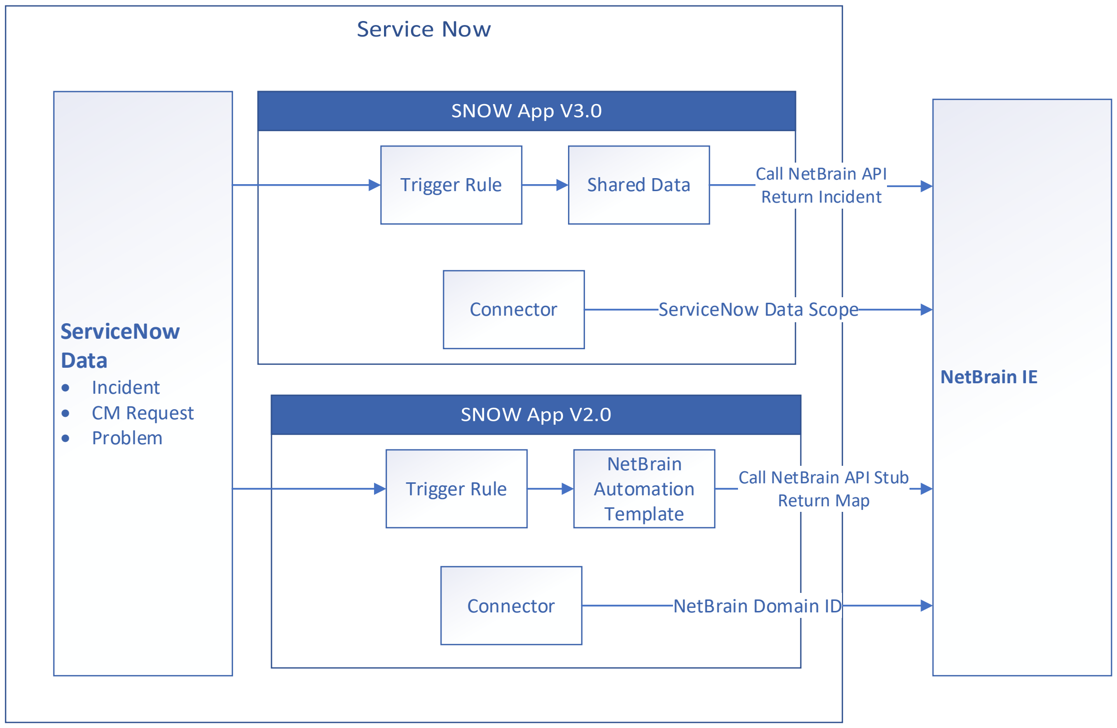
Display all triggered tasks from integrated IT system and NetBrain incident.

Items: 9 Time Range: Last 7 days Status: All Search...

Task ID	Source	Category	Triggered Time	Status	Incident Type	Incident ID	Matched Diagnosis Count	Log
b67c1a1e-1311-4d8a-a124-a946b12b9e35	ServiceNow	incident	1/17/2022 01:53 PM	Finished	Demo_IT	100007	0	View
8489e78b-9664-49c4-8e69-09ca1394f221	ServiceNow	incident	1/17/2022 01:38 PM	Finished	Demo_IT	100001		View
6a09801f-6b1a-ce58-3acf-08af34e09449	NetBrain	--	1/17/2022 10:27 AM	Finished	None	100006	1	View
93ba5a9c-b7ff-43a6-9bc3-9534d3cac955	ServiceNow	incident	1/16/2022 09:44 AM	Finished	Demo_IT	100005	1	View
27dcb02b-af9b-41ca-8d6f-41e3c01b4e54	ServiceNow	incident	1/16/2022 09:42 AM	Finished	Demo_IT	100004	1	View
a1b08f70-bd2a-4ad5-9956-5c8c29e83cb2	ServiceNow	incident	1/16/2022 09:25 AM	Finished	Demo_IT	100003	1	View
15a9f304-93a9-4b73-a88f-04a4ebdfc1b6	ServiceNow	incident	1/16/2022 09:21 AM	Finished	Demo_IT	100002	1	View
a5300e1e-5952-45b3-8b95-48a899387479	ServiceNow	incident	1/16/2022 09:09 AM	Finished	Demo_IT	100001	1	View
8e96c8b0-7dfb-4258-981d-631e5c734941	ServiceNow	incident	1/16/2022 09:09 AM	Failed	Demo_IT	None	0	View

2.3 ServiceNow App 3.0

IE Release 10.0 and 10.1 provide new features to improve the automation troubleshooting workflow, such as Incident Portal, Guidebook, Network Intention (NI), and Network Intention Cluster (NIC). To integrate these new features into the workflow, NetBrain v10.1 developed a new Triggered Automation Framework (TAF) and a new version of the ServiceNow App, 3.0. The ServiceNow App 3.0 (abbreviated as App 3.0) has the following improvements:



- Move the definition of NetBrain automation from the App to the NetBrain IE system and so do not require the App definition change to automate a new network task.

  App 2.0 defines the trigger rule and automation template for each type of network problem. When a new task issue needs to be automated, or better automation is developed for an existing task issue, the automation template will be redefined, which requires ServiceNow admin to be involved. The new TAF framework moves the definition of the automation template to the IE system. So, App 3.0 only needs to send the ServiceNow data helpful in diagnosing to the IE system. Since a ServiceNow data resource such as the Incident table does not change, App 3.0 does not require reconfiguration when a new type of problem occurs.

- Support what data will be sent to NetBrain IE

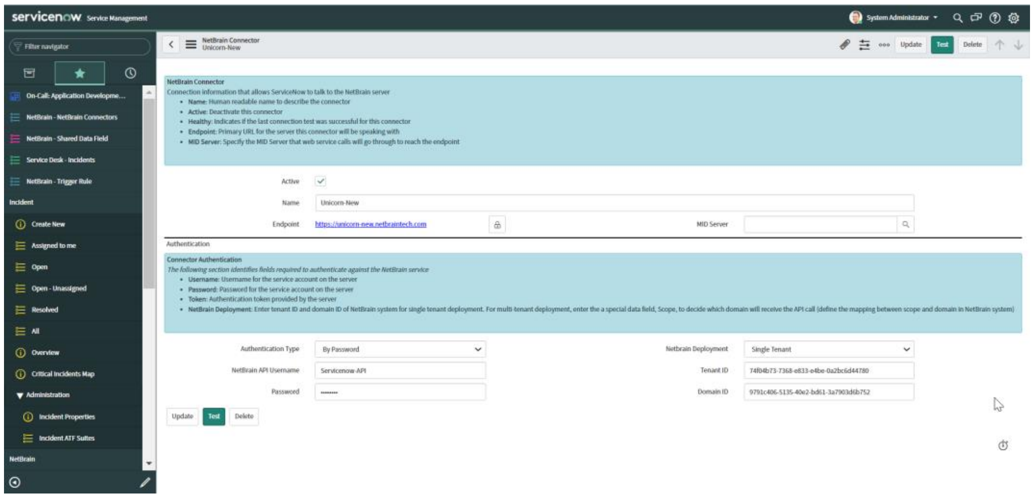
App3.0 completely upgrades problem diagnosis automation and mapping engine from fixed, runbook-driven to a flexible, intent-driven approach, enabling ServiceNow as another independent source to consume NetBrain automation. It also makes it possible to upgrade the ServiceNow/NetBrain integration without coding continuously.

- Integrate with NetBrain incident and display the NI and NIC results
  When triggered automation occurs, the NetBrain incident URL will be returned to ServiceNow to replace the current Embed Map. A user can open NetBrain Incident Pane from the triggered results, which provides much richer data and diagnosis history than the static embed map. Also, this updated ServiceNow data entry can be sent to NetBrain IE and displayed in the Incident Pane. In addition, the status code of NI and NIC results are displayed in the triggered results.

- Support Multi-tenant
  The MSP customers can have multiple tenants or domains to manage their clients' networks. In App 2.0, we only support one domain for triggered automation. App 3.0 supports the multi-tenant by mapping the related ServiceNow data (Scope) to NetBrain tenants and domains.

2.3.1 NetBrain Connector

The NetBrain Connector connects the ServiceNow and the NetBrain IE system. App 3.0 supports a new authentication method, by token, and the multi-tenant deployment.



NetBrain Connector
Connection information that allows ServiceNow to talk to the NetBrain server
- **Name:** Human readable name to describe the connector
- **Active:** Deactivate this connector
- **Healthy:** Indicates if the last connection test was successful for this connector
- **Endpoint:** Primary URL for the server this connector will be speaking with
- **MID Server:** Specify the MID Server that web service calls will go through to reach the endpoint

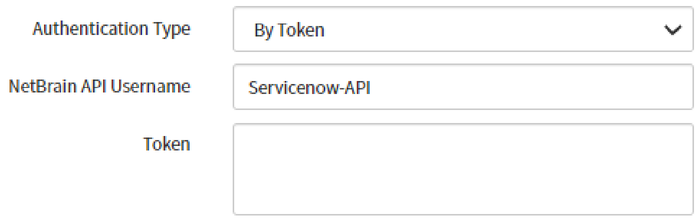
Active	✓
Name	Unicorn-New
Endpoint	https://unicorn-new.netbraintech.com
MID Server	

Authentication

Connector Authentication
The following section identifies fields required to authenticate against the NetBrain service
- **Username:** Username for the service account on the server
- **Password:** Password for the service account on the server
- **Token:** Authentication token provided by the server
- **NetBrain Deployment:** Enter tenant ID and domain ID of NetBrain system for single tenant deployment. For multi-tenant deployment, enter the a special data field, Scope, to decide which domain will receive the API call (define the mapping between scope and domain in NetBrain system)

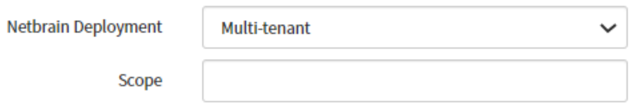
Authentication Type	By Password
NetBrain API Username	Servicenow-API
Password	••••••••
Netbrain Deployment	Single Tenant
Tenant ID	74f04b73-7368-e833-e4be-0a2bc6d44780
Domain ID	9791c406-5135-40e2-bd61-3a7903d6b752

Besides the old authentication by NetBrain API username and password, the App 3.0 supports the authentication by API username and token, which can be used if NetBrain IE user uses token for authentication.



Authentication Type	By Token
NetBrain API Username	Servicenow-API
Token	

MSP customers usually have a multiple-tenant system, one tenant for one client. App 3.0 Connectors support the multiple tenants via the concept of Scope. The Scope is used to match a triggered API call to a NetBrain tenant. The Scope should be the display name (field name) identifying which client a ServiceNow ticket belongs to, such as ***company name*** *and* ***customer ID***.



Netbrain Deployment	Multi-tenant
Scope	

The mappings between scopes and domains are defined in the NetBrain IE system setting.

2.3.2 Define Shared Data Field

The App 3.0 allows the user to define the data fields sent from ServiceNow to NetBrain IE system in the API call triggered. The shared data can be defined for each source table, like ***Incident***, ***Problem***, ***Change Request***, or other customized tables.

All data fields are displayed by default while defining the shared data for a ServiceNow source. The Filter function is provided for the user to find the fields. The default filter lists this source's matched fields and key subfields. The App also provides the other type of filter to list all subfields of a matched field, for example, ***company.class***, etc.

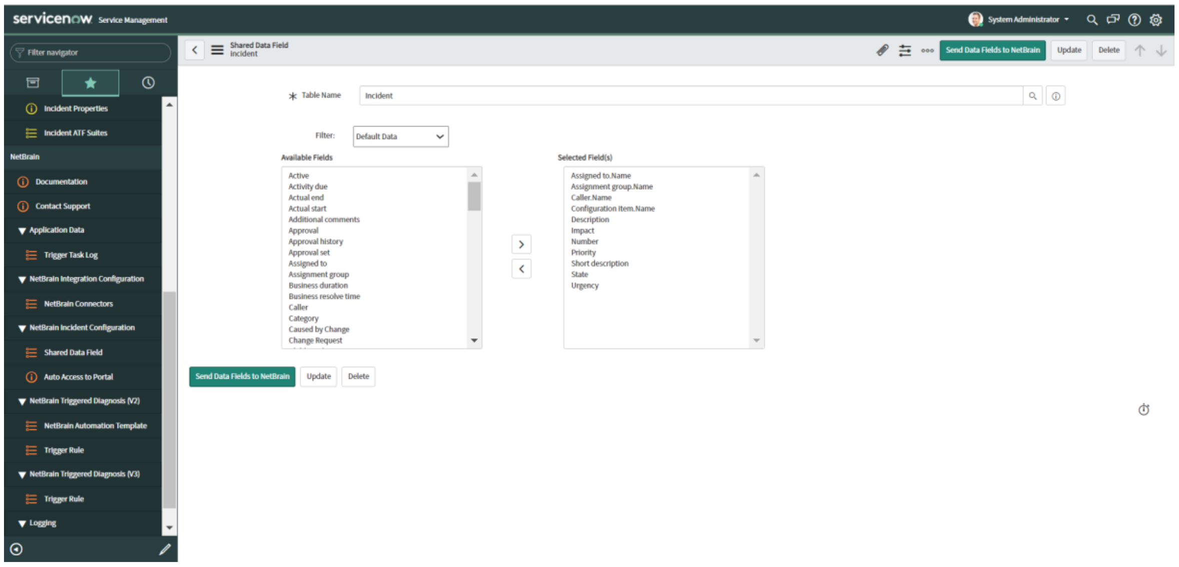


Table Name: Incident

Filter: Default Data

Available Fields: Active, Activity due, Actual end, Actual start, Additional comments, Approval, Approval history, Approval set, Assigned to, Assignment group, Business duration, Business resolve time, Caller, Category, Caused by Change, Change Request

Selected Field(s): Assigned to.Name, Assignment group.Name, Caller.Name, Configuration item.Name, Description, Impact, Number, Priority, Short description, State, Urgency

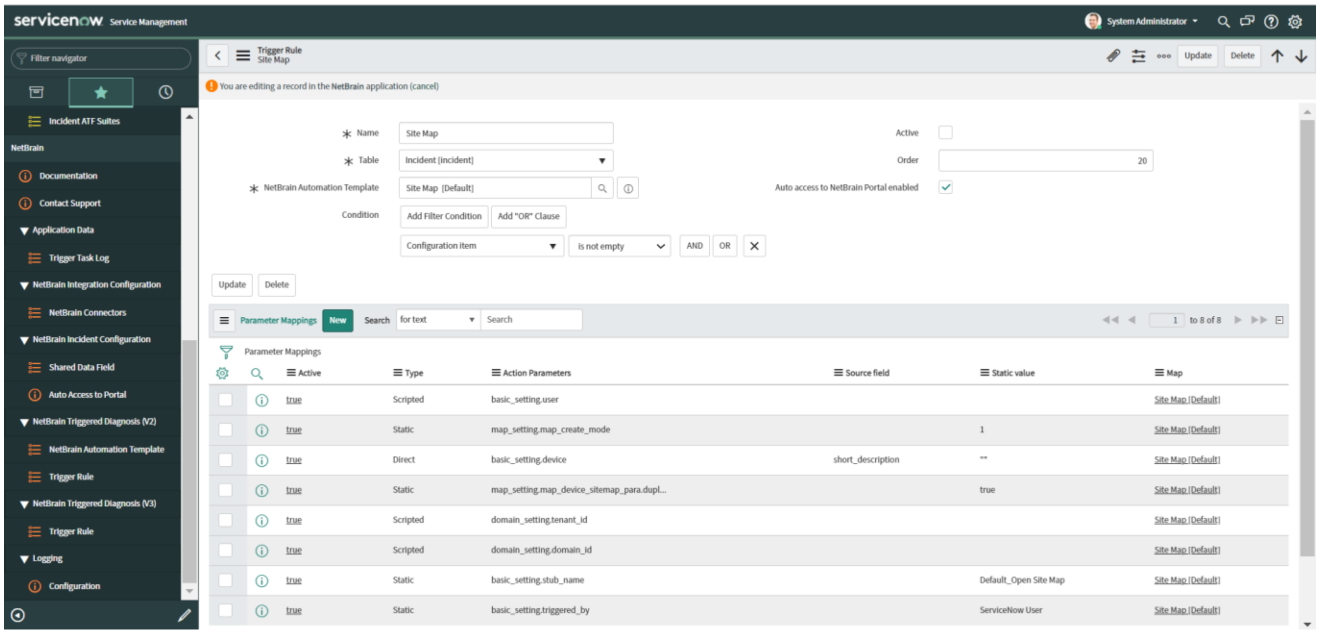
2.3.3 Auto Trigger for NetBrain Problem Diagnosis

App 3.0 supports two types of triggers:

- Diagnosis Trigger V2.0: identical to the functions of App 2.0. This stub will create a NetBrain map and execute the Runbook.
- Diagnosis Trigger V3.0: the new diagnosis trigger supports multiple tenants and domains. It will trigger Network Intent Cluster (NIC) and Network Intent (NI).

2.3.3.1 Trigger Stub V2

App 2.0 provided trigger rule and NetBrain automation template to create NetBrain map and execute a Runbook. App 3.0 keeps these functions under NetBrain Triggered Diagnosis (V2) to be compatible with App 2.0 with one enhancement: the user has ***Enable Auto Access for NetBrain Portal*** option in trigger rule setting. With this option checked, the NetBrain Portal of the Incident will create an access code, and the user can open the Incident Portal without login.



You are editing a record in the NetBrain application (cancel)

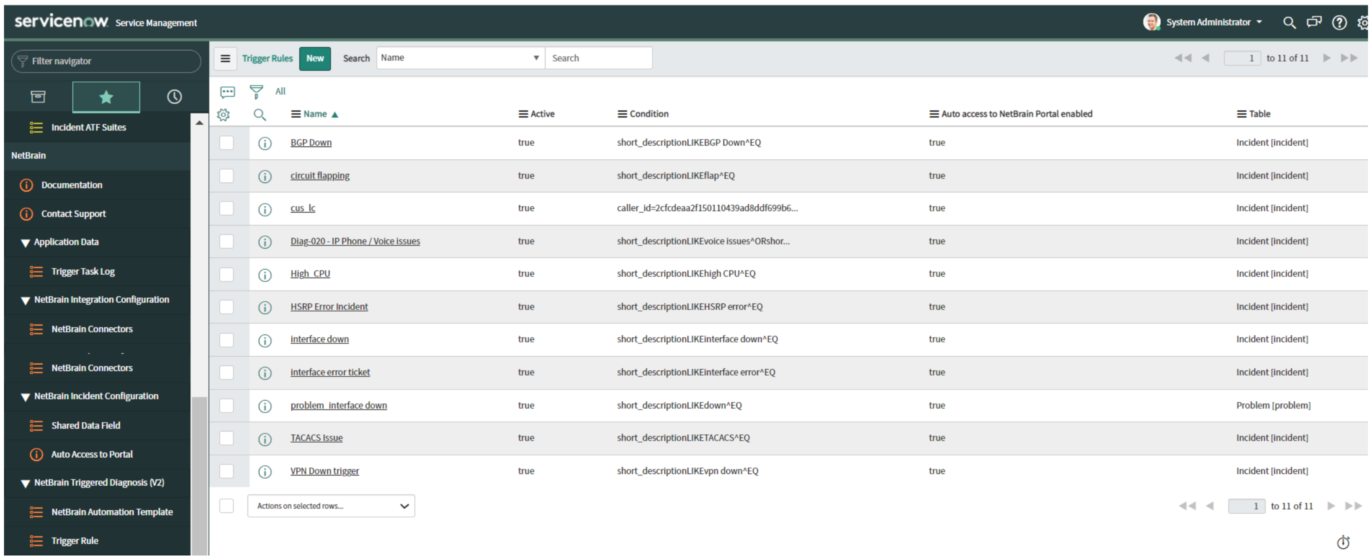
Name	Site Map	Active	☐
Table	Incident [incident]	Order	20
NetBrain Automation Template	Site Map [Default]	Auto access to NetBrain Portal enabled	✓
Condition	Configuration item / is not empty		

Parameter Mappings — 1 to 8 of 8

Active	Type	Action Parameters	Source field	Static value	Map
true	Scripted	basic_setting.user			Site Map [Default]
true	Static	map_setting.map_create_mode		1	Site Map [Default]
true	Direct	basic_setting.device	short_description	""	Site Map [Default]
true	Static	map_setting.map_device_sitemap_para.dupl...		true	Site Map [Default]
true	Scripted	domain_setting.tenant_id			Site Map [Default]
true	Scripted	domain_setting.domain_id			Site Map [Default]
true	Static	basic_setting.stub_name		Default_Open Site Map	Site Map [Default]
true	Static	basic_setting.triggered_by		ServiceNow User	Site Map [Default]

2.3.3.2 Trigger Diagnosis V3

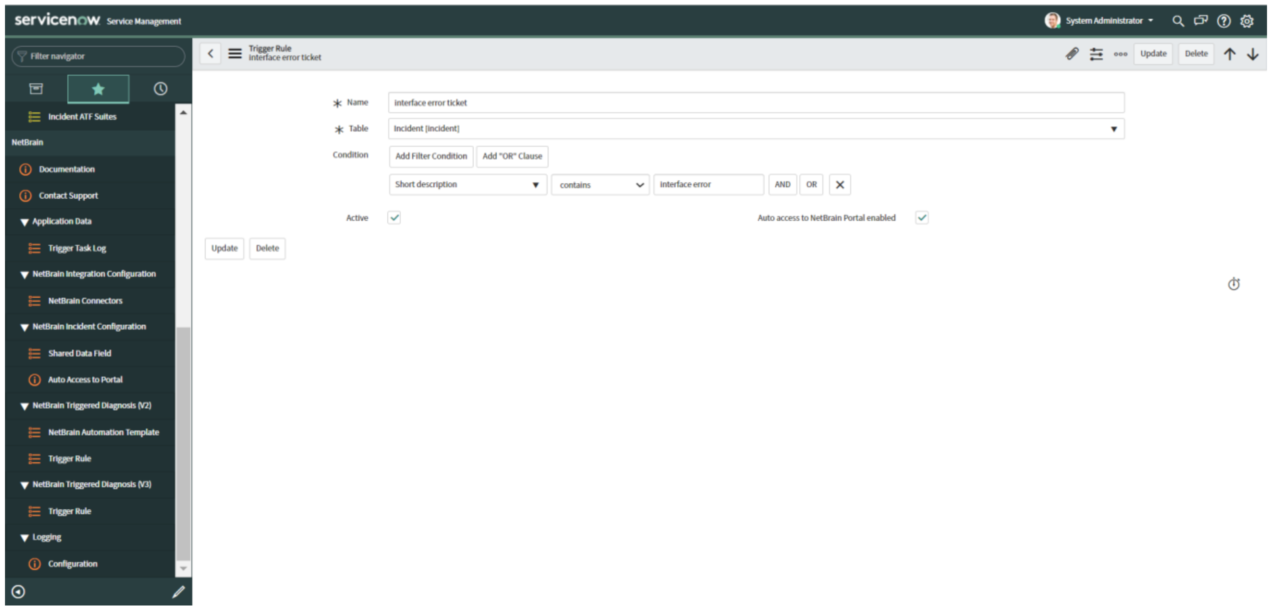
NetBrain Triggered Diagnosis V3 can trigger NIC and NI.



Trigger Rules — 1 to 11 of 11

Name	Active	Condition	Auto access to NetBrain Portal enabled	Table
BGP Down	true	short_descriptionLIKEBGP Down^EQ	true	Incident [incident]
circuit flapping	true	short_descriptionLIKEflap^EQ	true	Incident [incident]
cus_lc	true	caller_id=2cfcdeaa2f150110439ad8ddf699b6...	true	Incident [incident]
Diag-020 - IP Phone / Voice issues	true	short_descriptionLIKEvoice issues^ORshor...	true	Incident [incident]
High_CPU	true	short_descriptionLIKEhigh CPU^EQ	true	Incident [incident]
HSRP Error Incident	true	short_descriptionLIKEHSRP error^EQ	true	Incident [incident]
interface down	true	short_descriptionLIKEinterface down^EQ	true	Incident [incident]
interface error ticket	true	short_descriptionLIKEinterface error^EQ	true	Incident [incident]
problem_interface down	true	short_descriptionLIKEdown^EQ	true	Problem [problem]
TACACS Issue	true	short_descriptionLIKETACACS^EQ	true	Incident [incident]
VPN Down trigger	true	short_descriptionLIKEvpn down^EQ	true	Incident [incident]

The definition of the trigger rule of Triggered Diagnosis V3 is much simpler than V2. The user only needs to define the source table and conditions in which an API call is triggered. So, App 3.0 only needs to send the ServiceNow shared data defined in section 3 to the IE system for the IE system to define how the data is mapped to NetBrain data and what automation tasks are to be executed.



Name: interface error ticket

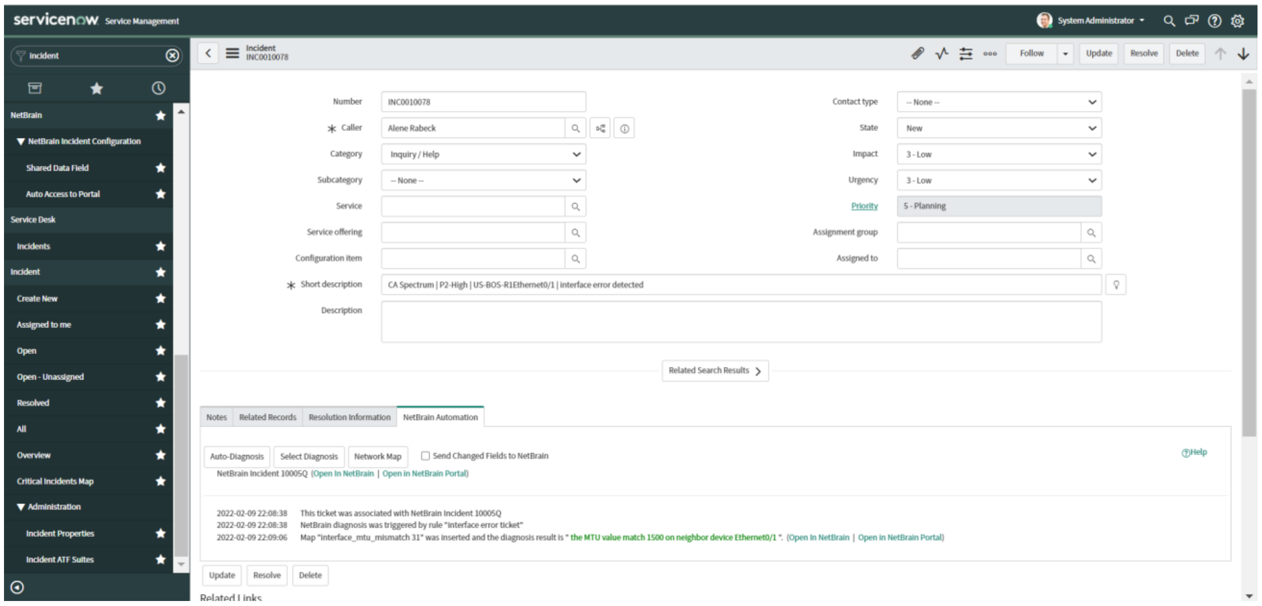
Table: Incident [incident]

Condition: Short description | contains | interface error

Active ✓ Auto access to NetBrain Portal enabled ✓

2.3.4 Auto Triggered Results

The triggered logs and results are displayed under the **NetBrain Automation** tab of the ServiceNow ticket. The logs show each step of the triggered events with the timestamps, such as creating an incident, creating a map, executing a Runbook, executing a NIC.



Incident INC0010078

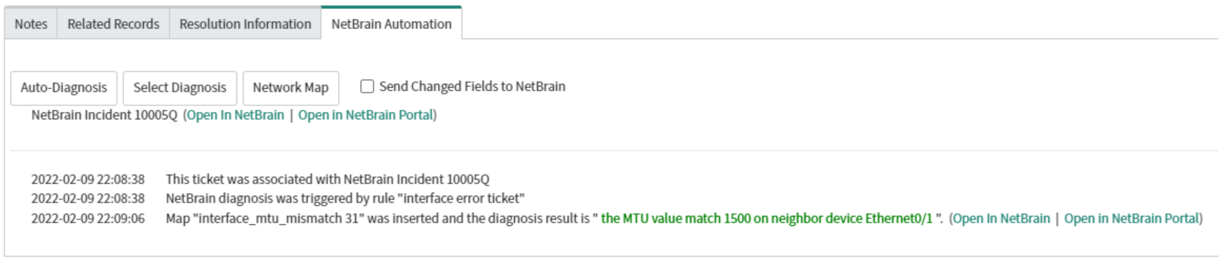
Number	INC0010078	Contact type	-- None --
Caller	Alene Rabeck	State	New
Category	Inquiry / Help	Impact	3 - Low
Subcategory	-- None --	Urgency	3 - Low
Service		Priority	5 - Planning
Service offering		Assignment group	
Configuration item		Assigned to	
Short description	CA Spectrum | P2-High | US-BOS-R1Ethernet0/1 | interface error detected		
Description			

Send Changed Fields to NetBrain
NetBrain Incident 10005Q (Open In NetBrain | Open in NetBrain Portal)

```
2022-02-09 22:08:38   This ticket was associated with NetBrain Incident 10005Q
2022-02-09 22:08:38   NetBrain diagnosis was triggered by rule "interface error ticket"
2022-02-09 22:09:06   Map "interface_mtu_mismatch 31" was inserted and the diagnosis result is " the MTU value match 1500 on neighbor device Ethernet0/1 ". (Open In NetBrain | Open in NetBrain Portal)
```

There are two improvements:

- In App 3.0, we recommend viewing the triggered results via the NetBrain Incident Pane instead of the embed map. The Incident Pane provides much richer data and diagnosis history than the static embed map. The App 3.0 provides the links for a user to open the Incident map in the NetBrain IE system or Portal, which includes all troubleshooting information.



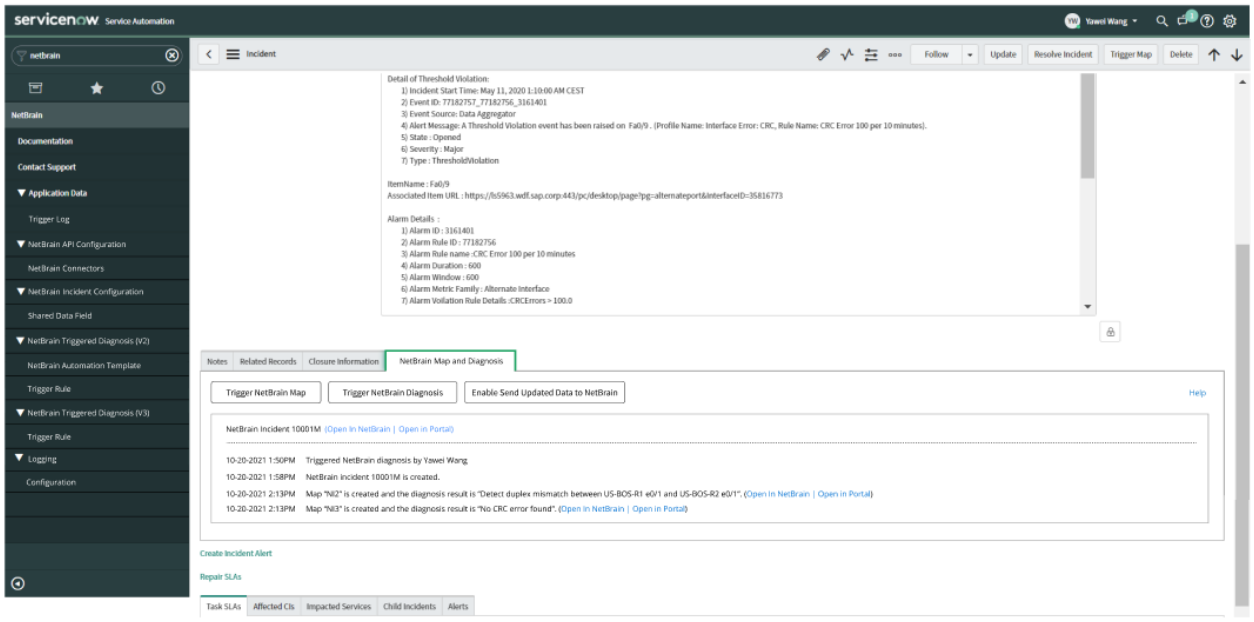
Send Changed Fields to NetBrain
NetBrain Incident 10005Q (Open In NetBrain | Open in NetBrain Portal)

```
2022-02-09 22:08:38   This ticket was associated with NetBrain Incident 10005Q
2022-02-09 22:08:38   NetBrain diagnosis was triggered by rule "interface error ticket"
2022-02-09 22:09:06   Map "interface_mtu_mismatch 31" was inserted and the diagnosis result is " the MTU value match 1500 on neighbor device Ethernet0/1 ". (Open In NetBrain | Open in NetBrain Portal)
```

- The NIC or NI diagnosis status is displayed on the ServiceNow ticket page.

2.3.5 Manual Trigger NetBrain Map and Diagnosis

App 3.0 allows users to manually trigger the NetBrain map and Diagnosis from the triggered results.



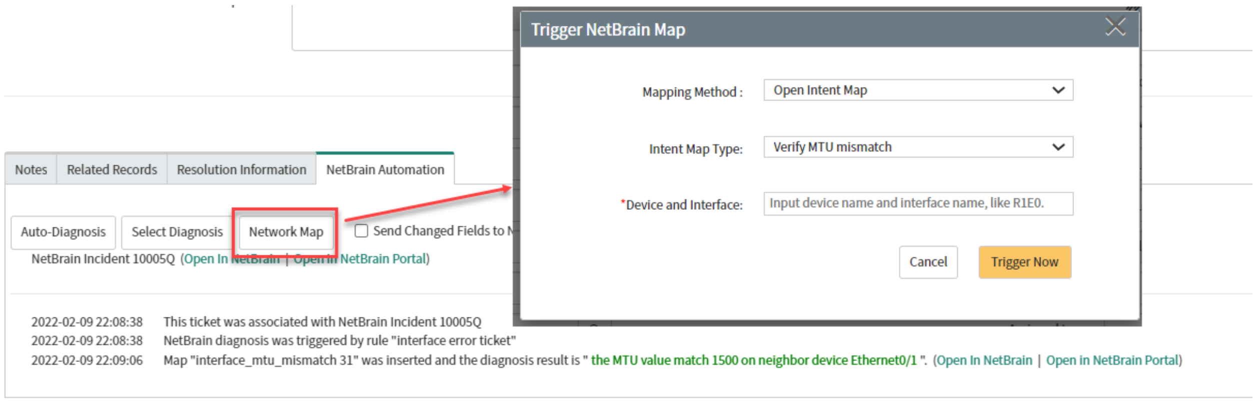
```
Detail of Threshold Violation:
    1) Incident Start Time: May 11, 2020 1:10:00 AM CEST
    2) Event ID: 77182757_77182756_3161401
    3) Event Source: Data Aggregator
    4) Alert Message: A Threshold Violation event has been raised on  Fa0/9 . (Profile Name: Interface Error: CRC, Rule Name: CRC Error 100 per 10 minutes).
    5) State : Opened
    6) Severity : Major
    7) Type : ThresholdViolation

ItemName : Fa0/9
Associated Item URL : https://ls5963.wdf.sap.corp:443/pc/desktop/page?pg=alternateport&InterfaceID=35816773

Alarm Details :
    1) Alarm ID : 3161401
    2) Alarm Rule ID : 77182756
    3) Alarm Rule name :CRC Error 100 per 10 minutes
    4) Alarm Duration : 600
    5) Alarm Window : 600
    6) Alarm Metric Family : Alternate Interface
    7) Alarm Voilation Rule Details :CRCErrors > 100.0
```

NetBrain Incident 10001M (Open In NetBrain | Open in Portal)

```
10-20-2021 1:50PM   Triggered NetBrain diagnosis by Yawei Wang
10-20-2021 1:58PM   NetBrain incident 10001M is created.
10-20-2021 2:13PM   Map "NI2" is created and the diagnosis result is "Detect duplex mismatch between US-BOS-R1 e0/1 and US-BOS-R2 e0/1". (Open In NetBrain | Open in Portal)
10-20-2021 2:13PM   Map "NI3" is created and the diagnosis result is "No CRC error found". (Open In NetBrain | Open in Portal)
```

2.3.5.1 Manually Trigger NetBrain Map

App 3.0 provides a user with four methods to create the map: **Map Device and Its Neighbor, map a path, open the site map,** and **intent map**. For each method, a user can define the input.



Mapping Method : Open Intent Map

Intent Map Type: Verify MTU mismatch

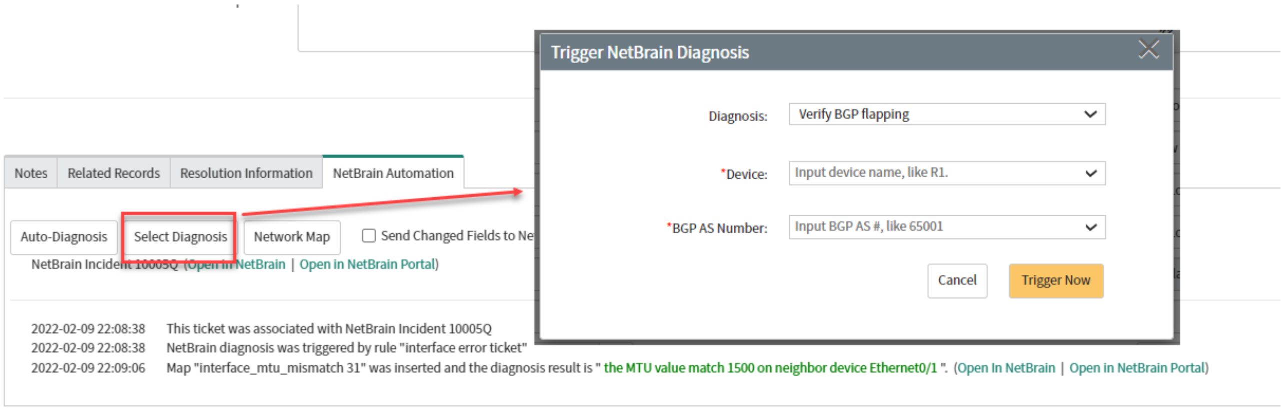
*Device and Interface: Input device name and interface name, like R1E0.

NetBrain Incident 10005Q (Open In NetBrain | Open in NetBrain Portal)

```
2022-02-09 22:08:38   This ticket was associated with NetBrain Incident 10005Q
2022-02-09 22:08:38   NetBrain diagnosis was triggered by rule "interface error ticket"
2022-02-09 22:09:06   Map "interface_mtu_mismatch 31" was inserted and the diagnosis result is " the MTU value match 1500 on neighbor device Ethernet0/1 ". (Open In NetBrain | Open in NetBrain Portal)
```

2.3.5.2 Trigger NetBrain Diagnosis

The manual Trigger NetBrain Diagnosis function provides a user to execute a NIC or NI. The available diagnosis list is generated by the IE system. The method may require the user to enter the input, which can be mandatory or optional. For example, a diagnosis related to BGP will require a BGP AS number and optionally one or multiple device names.



Diagnosis: Verify BGP flapping

*Device: Input device name, like R1.

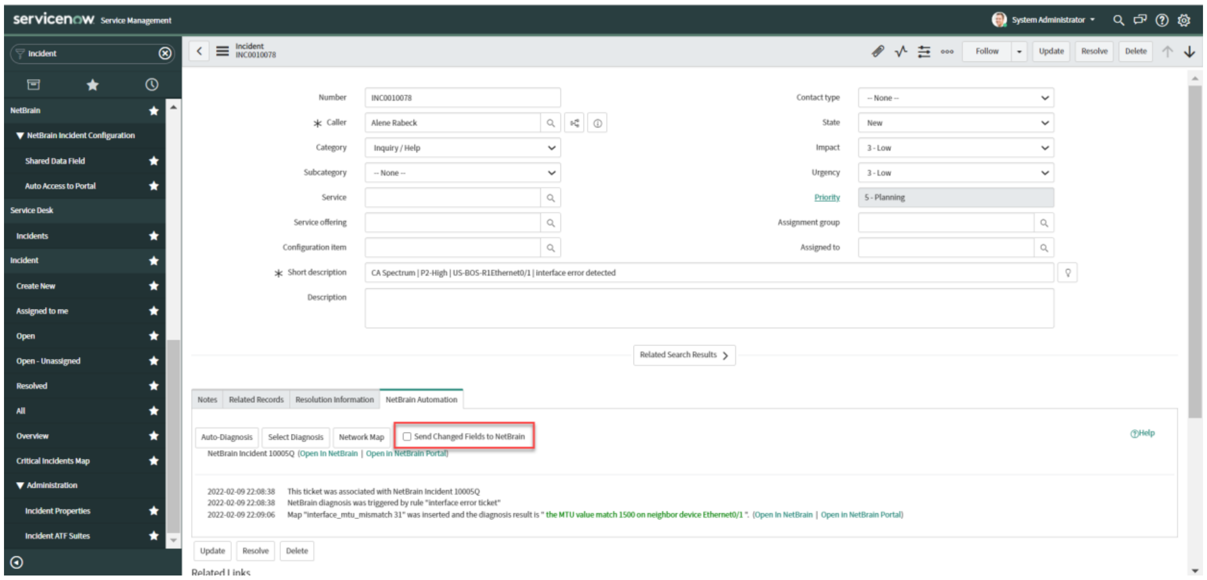
*BGP AS Number: Input BGP AS #, like 65001

NetBrain Incident 10005Q (Open in NetBrain | Open in NetBrain Portal)

```
2022-02-09 22:08:38   This ticket was associated with NetBrain Incident 10005Q
2022-02-09 22:08:38   NetBrain diagnosis was triggered by rule "interface error ticket"
2022-02-09 22:09:06   Map "interface_mtu_mismatch 31" was inserted and the diagnosis result is " the MTU value match 1500 on neighbor device Ethernet0/1 ".  (Open In NetBrain | Open in NetBrain Portal)
```

2.3.5.3 Enable Send Updated Data to NetBrain

If the value of the shared data changes, App 3.0 will not trigger a call to update the value by default. The option ***Send changed fields to NetBrain*** can enable the updated values to be sent to NetBrain. The updated values will be displayed in NetBrain Incident Portal.



Number	INC0010078	Contact type	-- None --
* Caller	Alene Rabeck	State	New
Category	Inquiry / Help	Impact	3 - Low
Subcategory	-- None --	Urgency	3 - Low
Service		Priority	5 - Planning
Service offering		Assignment group	
Configuration item		Assigned to	
* Short description	CA Spectrum | P2-High | US-BOS-R1Ethernet0/1 | interface error detected		
Description			

NetBrain Incident 10005Q (Open In NetBrain | Open in NetBrain Portal)

```
2022-02-09 22:08:38   This ticket was associated with NetBrain Incident 10005Q
2022-02-09 22:08:38   NetBrain diagnosis was triggered by rule "interface error ticket"
2022-02-09 22:09:06   Map "interface_mtu_mismatch 31" was inserted and the diagnosis result is " the MTU value match 1500 on neighbor device Ethernet0/1 ".  (Open In NetBrain | Open in NetBrain Portal)
```

2.4 Incident Pane Enhancements

The Incident is the Problem Diagnosis Automation System (PDAS) output, whose data will be displayed in the incident pane and portal. In v10.1, the system will create an incident for each ticket when triggered automation occurs. The user is redirected to the NetBrain incident pane from the customer ticket system, e.g., ServiceNow, which provides much richer data and diagnosis history than the static embedded map in ServiceNow App.

Incident Pane provides a central collaboration platform for troubleshooting and data sharing, including:

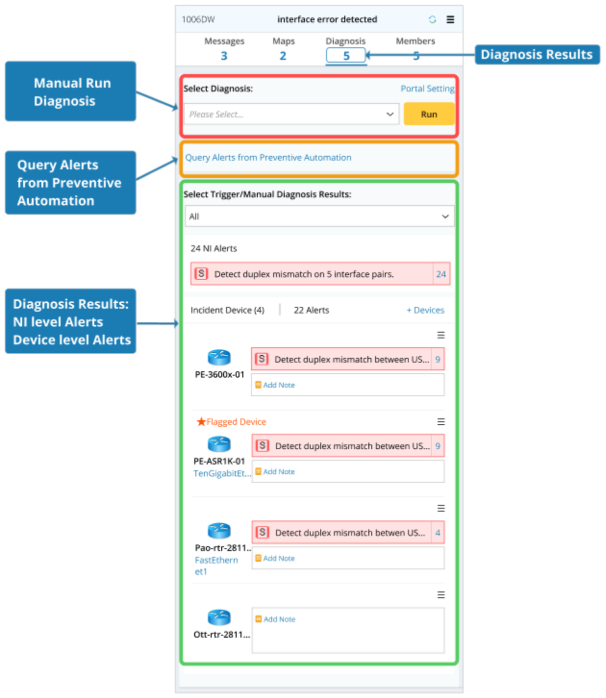
- External ticket information, such as ServiceNow ticket ID, short description, and call back URL.
- Problem area mappings
- NetBrain Flash alert from the adaptive monitoring, shown as the incident message
- Network Intent diagnosis result, shown in incident diagnosis tab
- User notes during collaborative troubleshooting
- Recommended guidebook and runbook template

V10.1 makes the following enhancements on the incident pane and incident portal.

- Display NI diagnosis results in incident pane (from TAF, Probe, manually run), making the incident pane the Single Pane of Glass (SPOG) of network Troubleshooting (TS).
- Enable the subscription of Adaptive Monitoring data (flash alert, NI status codes) to allow users to see related historical monitoring data in the incident pane.
- Allow users to run recommended diagnoses in the incident pane.

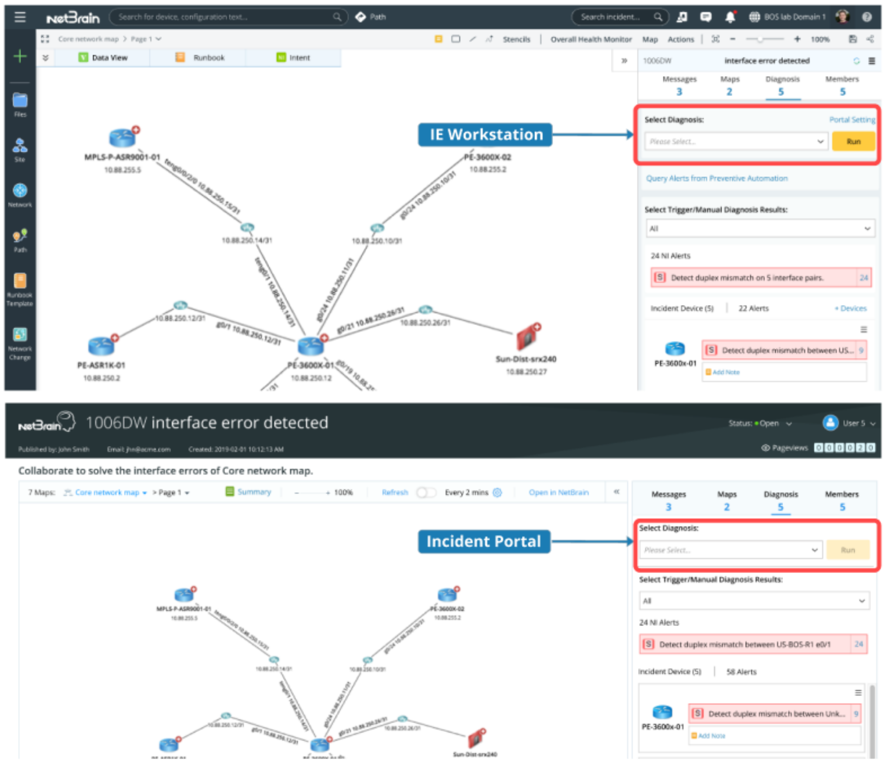
2.4.1 Browse Diagnosis Results and Run Diagnosis

The Incident pane has four tabs: Messages, Maps, Diagnosis, and Members. The Device tab of the earlier version is renamed the Diagnosis tab, under which Users can centrally view the diagnosis results from other functions and manually run the Diagnosis.


1006DW
interface error detected
Messages 3
Maps 2
Diagnosis 5
Members 5
Diagnosis Results
Manual Run Diagnosis
Select Diagnosis:
Portal Setting
Please Select...
Run
Query Alerts from Preventive Automation
Query Alerts from Preventive Automation
Select Trigger/Manual Diagnosis Results:
All
24 NI Alerts
Detect duplex mismatch on 5 interface pairs. 24
Diagnosis Results: NI level Alerts Device level Alerts
Incident Device (4)
22 Alerts
+ Devices
PE-3600x-01
Detect duplex mismatch between US... 9
Add Note
Flagged Device
PE-ASR1K-01
TenGigabitEt...
Detect duplex mismatch between US... 9
Add Note
Pao-rtr-2811...
FastEthernet1
Detect duplex mismatch betwen US... 4
Add Note
Ott-rtr-2811...
Add Note

- Manually execute NI.

Users can run self-service Diagnoses defined in TAF in IE Workstation and Portal. The execution results will be sent to the incident message and the output of the diagnosis pane.


IE Workstation
Incident Portal
Select Diagnosis:
Run
Query Alerts from Preventive Automation
Select Trigger/Manual Diagnosis Results:
24 NI Alerts
Detect duplex mismatch on 5 interface pairs.
1006DW interface error detected
Collaborate to solve the interface errors of Core network map.

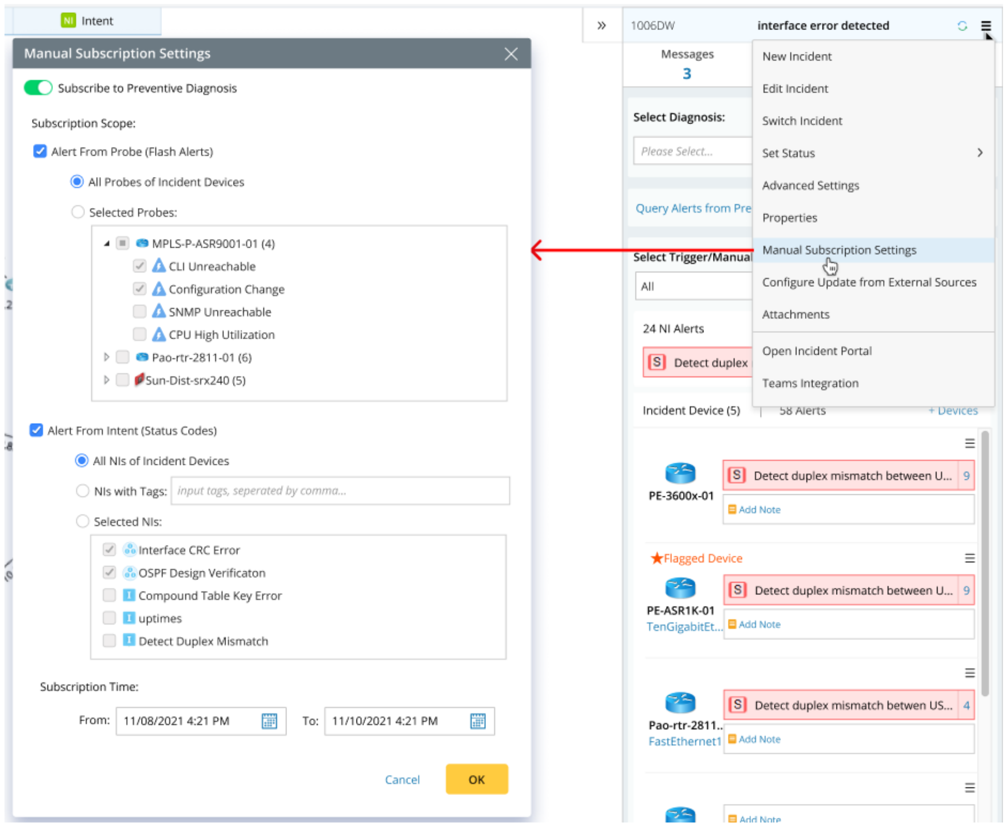
- Query Alerts from Preventive Automation

Users can query all alerts for incident devices or current map devices. A query summary is displayed in the Incent Pane, which links to Preventive Automation Dashboard.

2.4.2 Subscribe to Preventive Automation

Often solving a problem requires multi-person cooperation and various data types (such as map, NI, probe, Runbook...). Preventive Automation (Adaptive Monitoring) data subscription allows users to see all diagnosis results related to current network problems in the most recent time, which helps users locate and solve problems faster.

Users can choose which probes to subscribe to and which NI/NICs are included in the probe:



Manual Subscription Settings

Subscribe to Preventive Diagnosis

Subscription Scope:

- [x] Alert From Probe (Flash Alerts)
  - (●) All Probes of Incident Devices
  - ( ) Selected Probes:
    - MPLS-P-ASR9001-01 (4)
      - [x] CLI Unreachable
      - [x] Configuration Change
      - [ ] SNMP Unreachable
      - [ ] CPU High Utilization
    - Pao-rtr-2811-01 (6)
    - Sun-Dist-srx240 (5)
- [x] Alert From Intent (Status Codes)
  - (●) All NIs of Incident Devices
  - ( ) NIs with Tags: input tags, seperated by comma...
  - ( ) Selected NIs:
    - [x] Interface CRC Error
    - [x] OSPF Design Verificaton
    - [ ] Compound Table Key Error
    - [ ] uptimes
    - [ ] Detect Duplex Mismatch

Subscription Time:

From: 11/08/2021 4:21 PM To: 11/10/2021 4:21 PM

Cancel OK

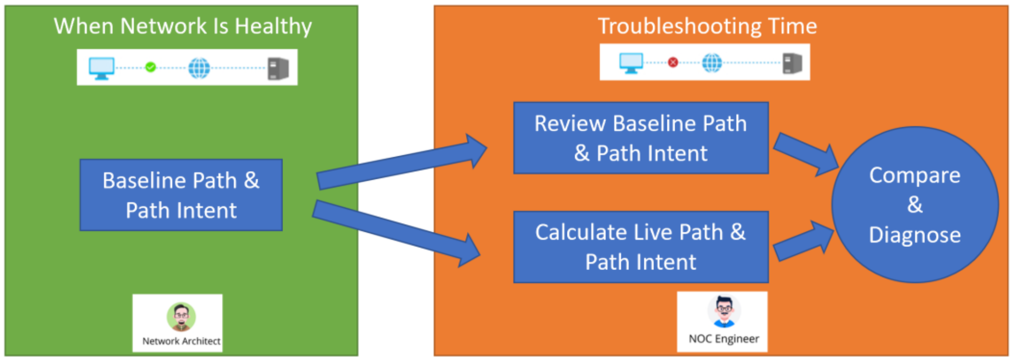
2.5 Path-based Troubleshooting Flow

The network is designed to carry critical application flows. A network is considered healthy if the critical applications flows are healthy, so **Path-based Troubleshooting Flow** (PBTF) is an essential part of PDAS, which intends to automate the Diagnosis of the repetitive problem and automate the enforcement of preventive measures (design rules, best practice, or security policy) across the entire network.

2.5.1 Upgraded PBTF

In previous versions, it was hard to define what a healthy application is like and diagnose the application slowness or do application impact analysis. V10.1 introduced the Path Intent feature and Intent-based new TAF and a few other Path-related function enhancements, which enable users to baseline, document, and define the diagnosis logic for the application path efficiently when the network is healthy. Then this well-documented and executable knowledge can help network engineers resolve application slowness issues more efficiently and do effective application impact analysis efficiently when a problem occurs on a network device or device interface.

The PBTF is upgraded in IEv10.1 as follows:


When Network Is Healthy
Baseline Path & Path Intent
Network Architect
Troubleshooting Time
Review Baseline Path & Path Intent
Calculate Live Path & Path Intent
Compare & Diagnose
NOC Engineer

Path-based Troubleshooting Flow in v10.1

1 Baseline Path and Path Intent, Add Path Intent into NIC (when network is healthy)

The critical application flows can be calculated via live network data with full documentation behind the path logic when a network is healthy. Path-related baseline data and diagnosis logic can be programmatically defined inside Path Intent without coding, which can be added into a NIC as static member NIs. Then, this NIC can be used in the trigger diagnosis of TAF with the filter of member NI defined with the path source and destination and the application name.

2 Trigger Path-Based Diagnosis (when the path issue occurs)

TAF receives the ticket sent by the 3rd IT system and triggers the execution of NIC according to the logic defined in TAF. If the ticket is related to an application, TAF will execute the path NIC, which will run those member NIs specified for this application. The alert message for this path will be displayed in the Incident pane.

3 Review Baseline Path and Path Intent (during troubleshooting)

During troubleshooting time, any user can intuitively access the pre-documented application path via the A|B path dialog. The pre-documented path results, along with the pre-built diagnosis automation, will accelerate the application troubleshooting process.

4 Calculate Live Path and Compare with Cached Path (during troubleshooting)

During troubleshooting, users can recalculate paths with the live data and compare them with the cached paths to reveal various issues behind a slow application.

5 Execute Path Intent and Diagnose Issues (during troubleshooting)

Users can rerun the associated path NI with the live network data and check the diagnosis results.

2.5.1.1 Main Target Problems and Use cases of PBTF

The enhanced Path-based Diagnosis functions intend to automate the following four kinds of typic path-based Diagnosis:

- **Is the Path changed visually?** Compare the cached path (when network is healthy), golden baseline path, and live Path in the troubleshooting stage.
- **Is the Path failing over programmatically?** Use NI to check routing table entry (for L3 failover) for the next-hop change and the CAM table (for L2 failover) change.
- **Is the Path healthy performance-wise?** Use NI to baseline and analyze the link utilization change, CPU/memory change, link error change, QoS buffer drop change, etc.
- **Is Path configured properly?** Use NI to check the QoS configuration consistency across devices of the path and configuration consistency between the failover device pair for the Path.

Two typical use cases can leverage the TAF and Path Intent-based diagnosis capabilities:

- Slow application analysis

  When an application performance monitoring finds an application slowness issue, it will create a ServiceNow ticket, and then NetBrain can be triggered by the ticket. TAF will filter and run the related Path Intents based on the application path information in the ticket (the path source/destination, application name, and path name) to find the root cause.

- Application impact analysis

When a network monitoring system finds issues occurred on a network device or device interface, it may create a ServiceNow ticket, which will trigger NetBrain Diagnosis. The Diagnosis can be configured to match and run all Path NIs associated with this device and report the applications impacted by this issue. The summary report will be displayed as an incident message.

2.5.2 PBTF Use Flow

There are two types of PBTF flows:

2.5.2.1 Flow 1: Path-based Interactive Troubleshooting Flow

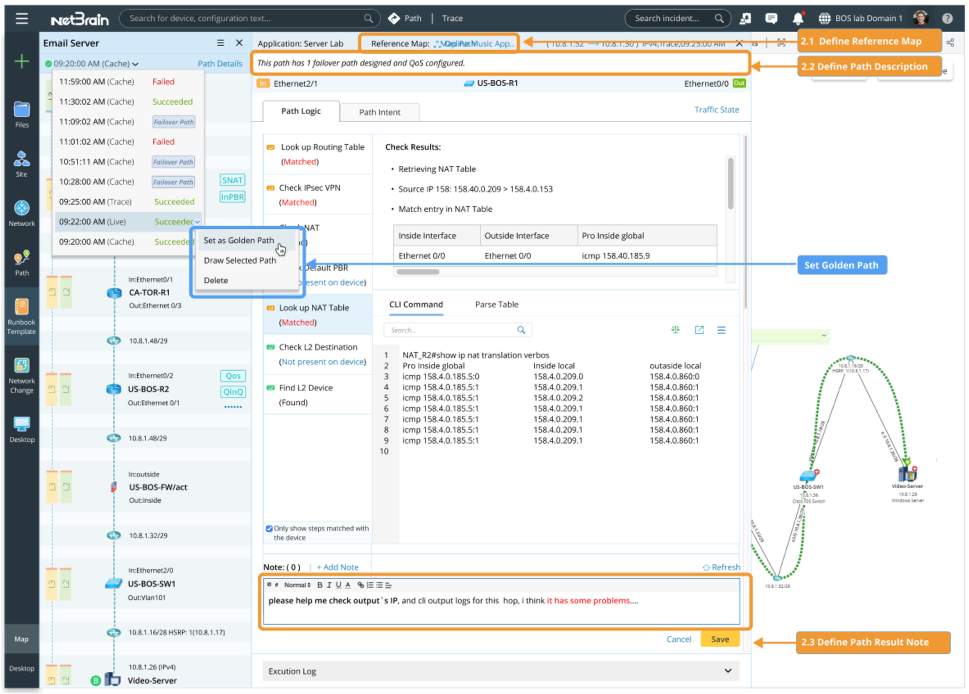
1. Define and discover path (When Network is Healthy)

The architect engineer can identify the critical applications and define the path source/destination and their gateways. Then, he can discover these paths when the network is healthy.

2. Document Path (When Network is Healthy)

V10.1 focuses on documenting the calculated paths and related path logics, including:

- Path description: like the business application name and description.
- Path note on each hop device to enter the related network design and troubleshooting knowledge.
- Path reference map: not just the map of path results, but also with design notes or related config-let, failover annotation, etc.
- Golden Path: the best traffic path as designed.



Email Server
09:20:00 AM (Cache)
Path Details
11:59:00 AM (Cache) Failed
11:30:02 AM (Cache) Succeeded
11:09:02 AM (Cache) Failover Path
11:01:02 AM (Cache) Failed
10:51:11 AM (Cache) Failover Path
10:28:00 AM (Cache) Failover Path
09:25:00 AM (Trace) Succeeded
09:22:00 AM (Live) Succeeded
09:20:00 AM (Cache) Succeeded
Set as Golden Path
Draw Selected Path
Delete
Application: Server Lab
Reference Map:
2.1 Define Reference Map
2.2 Define Path Description
Set Golden Path
2.3 Define Path Result Note
US-BOS-R1
Path Logic
Path Intent
Traffic State
CLI Command
Parse Table
Cancel
Save
Excution Log

This path has 1 failover path designed and QoS configured.

Look up Routing Table (Matched)
Check IPsec VPN (Matched)
Look up NAT Table (Matched)
Check L2 Destination (Not present on device)
Find L2 Device (Found)

Check Results:

- Retrieving NAT Table
- Source IP 158: 158.40.0.209 > 158.4.0.153
- Match entry in NAT Table

Inside Interface	Outside Interface	Pro Inside global
Ethernet 0/0	Ethernet 0/0	icmp 158.40.185.9

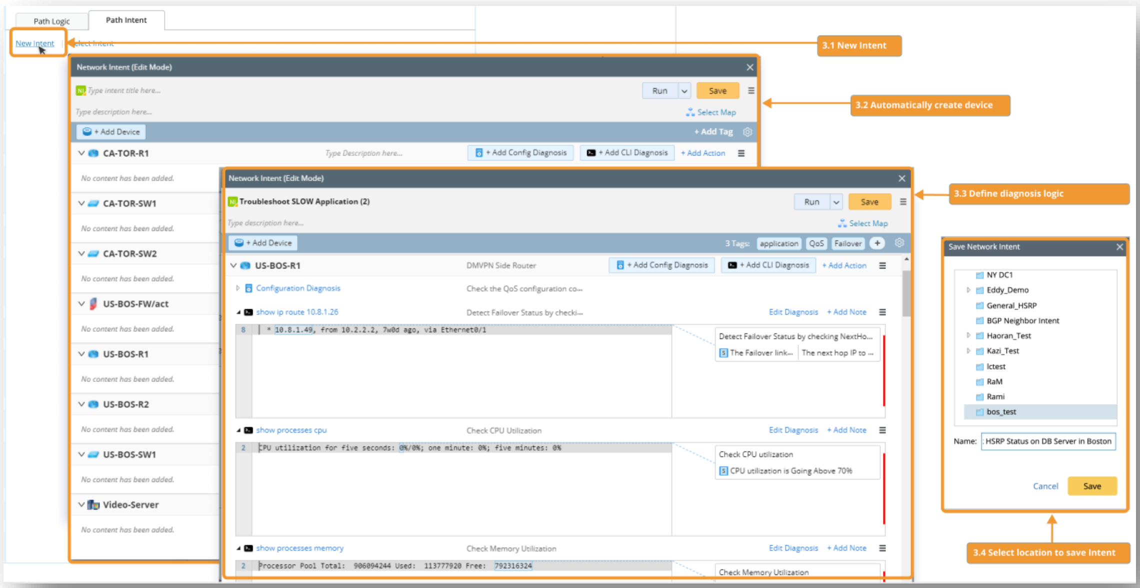
```
NAT_R2#show ip nat translation verbos
Pro Inside global       Inside local       outaside local
icmp 158.4.0.185.5:0    158.4.0.209.0      158.4.0.860:0
icmp 158.4.0.185.5:1    158.4.0.209.1      158.4.0.860:1
icmp 158.4.0.185.5:1    158.4.0.209.2      158.4.0.860:1
icmp 158.4.0.185.5:1    158.4.0.209.1      158.4.0.860:1
icmp 158.4.0.185.5:1    158.4.0.209.1      158.4.0.860:1
icmp 158.4.0.185.5:1    158.4.0.209.1      158.4.0.860:1
icmp 158.4.0.185.5:1    158.4.0.209.1      158.4.0.860:1
```

Note: (0) | + Add Note

please help me check output`s IP, and cli output logs for this hop, i think it has some problems....

3. Define Path NI (When Network is Healthy)

A network architect can define NI to document the path diagnosis logic when the network is healthy and associate it with the Path. When users define NI for a path, all hop devices of the Path will be put into NI. The path NI can diagnose slow application issues or other path issues.



```
* 10.8.1.49, from 10.2.2.2, 7w0d ago, via Ethernet0/1
```

```
CPU utilization for five seconds: 0%/0%; one minute: 0%; five minutes: 0%
```

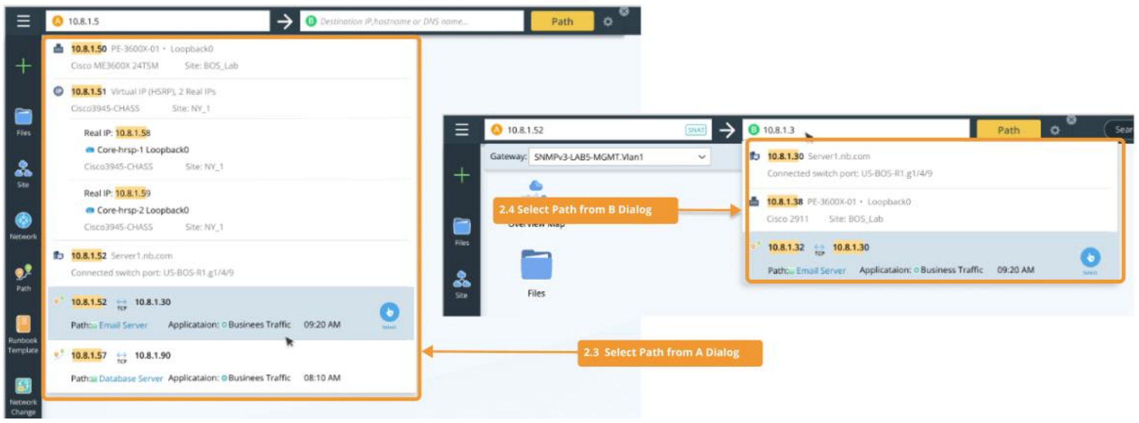
```
Processor Pool Total:  906094244 Used:  113777920 Free:  792316324
```

Typical examples of the Path NI:

- Check routing change for L3 failover
- Baseline and analyze the QoS buffer drop change

4. Find Pre-documented Path and Review Path Documents (During troubleshooting)

When a user enters the source and destination, the documented Path with the same source/destination will be displayed in the path IntelliSense dialog. The user can select one path to view its results on the map, including the path description, note, path reference map, and Path logic.


10.8.1.5
Path
10.8.1.50 PE-3600X-01 • Loopback0
10.8.1.51 Virtual IP (HSRP), 2 Real IPs
10.8.1.52 Server1.nb.com
10.8.1.52 ↔ 10.8.1.30
Path: Email Server Applicataion: Businees Traffic 09:20 AM
10.8.1.57 ↔ 10.8.1.90
Path: Database Server Applicataion: Businees Traffic 08:10 AM
10.8.1.52
10.8.1.3
Gateway: SNMPv3-LAB5-MGMT.Vlan1
10.8.1.30 Server1.nb.com
10.8.1.38 PE-3600X-01 • Loopback0
10.8.1.32 ↔ 10.8.1.30
2.4 Select Path from B Dialog
2.3 Select Path from A Dialog

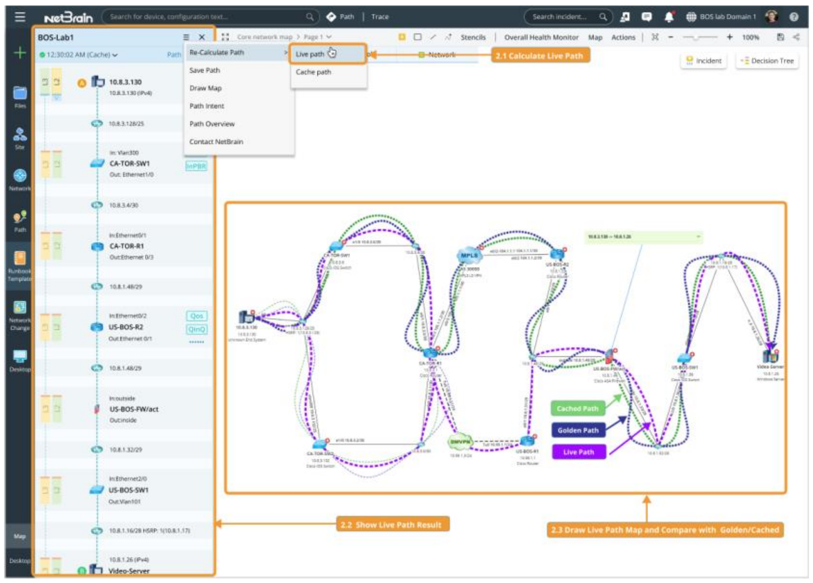
5. Drill-down analysis: calculate live Path and Compare with Cached Path (During troubleshooting)

During troubleshooting, users can discover the live path and compare it with the cached path.

a) Select golden or cached Path Results to draw it on the map.

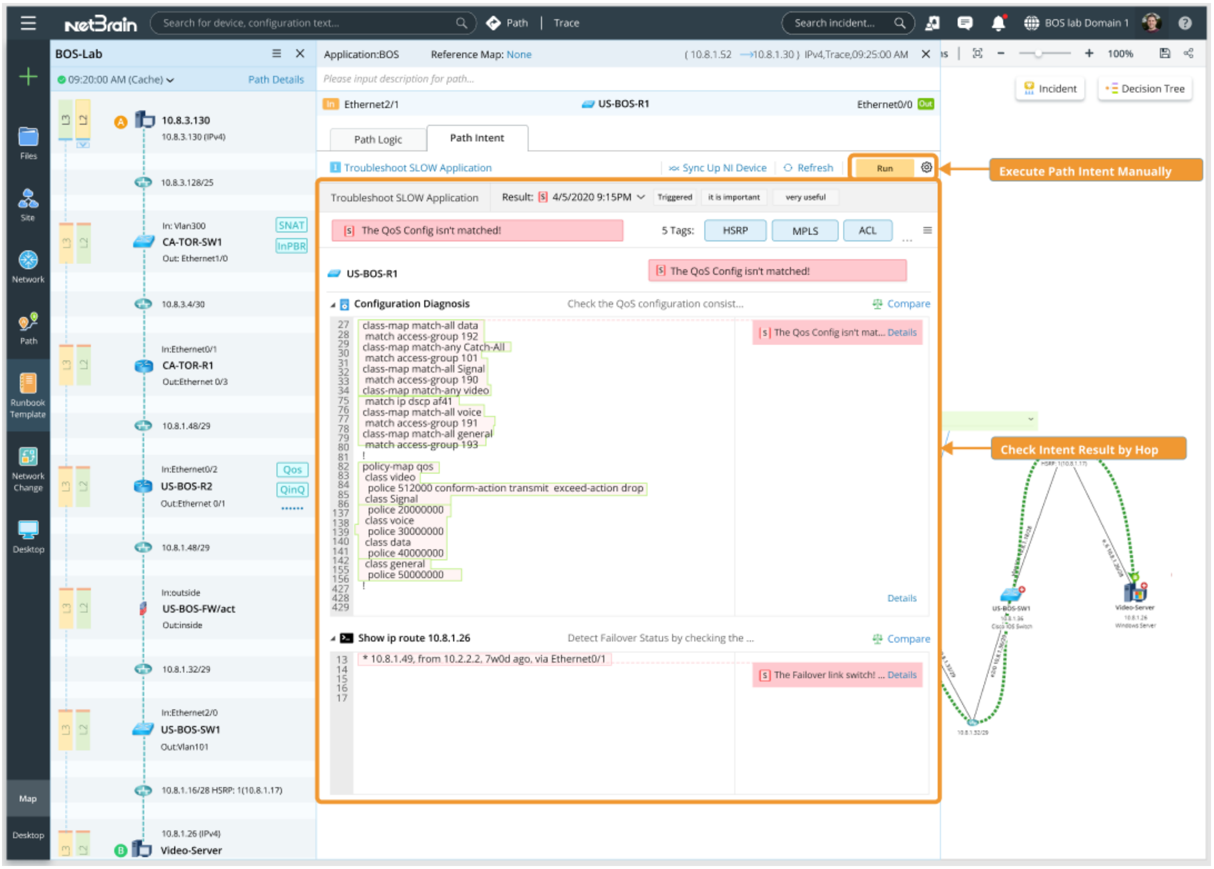
b) Calculate Live Path and compare it with golden Path or cached Path

If the live Path is different from the cached Path when the network is healthy, failover may have occurred to cause the application slowness due to the limited bandwidth of the backup link.



6. **Drill-down analysis - Execute Path intent (During troubleshooting)**

The user can browse alert summary messages in the Incident pane created by Trigger Diagnosis and open the corresponding Path NI to view the detailed diagnosis alerts. He can also run a Path NI to check the newest diagnosis results.



The QoS Config isn't matched!

US-BOS-R1

Configuration Diagnosis — Check the QoS configuration consist...

```
class-map match-all data
 match access-group 192
class-map match-any Catch-All
 match access-group 101
class-map match-all Signal
 match access-group 190
class-map match-any video
 match ip dscp af41
class-map match-all voice
 match access-group 191
class-map match-all general
 match access-group 193
!
policy-map qos
 class video
  police 512000 conform-action transmit  exceed-action drop
 class Signal
  police 20000000
 class voice
  police 30000000
 class data
  police 40000000
 class general
  police 50000000
!
```

The Qos Config isn't mat... Details

Show ip route 10.8.1.26 — Detect Failover Status by checking the ...

```
* 10.8.1.49, from 10.2.2.2, 7w0d ago, via Ethernet0/1
```

The Failover link switch! ... Details

2.5.2.2 Flow 2. Path-based Trigger Automation Flow

1-3. The first three steps are the same as Flow 1.

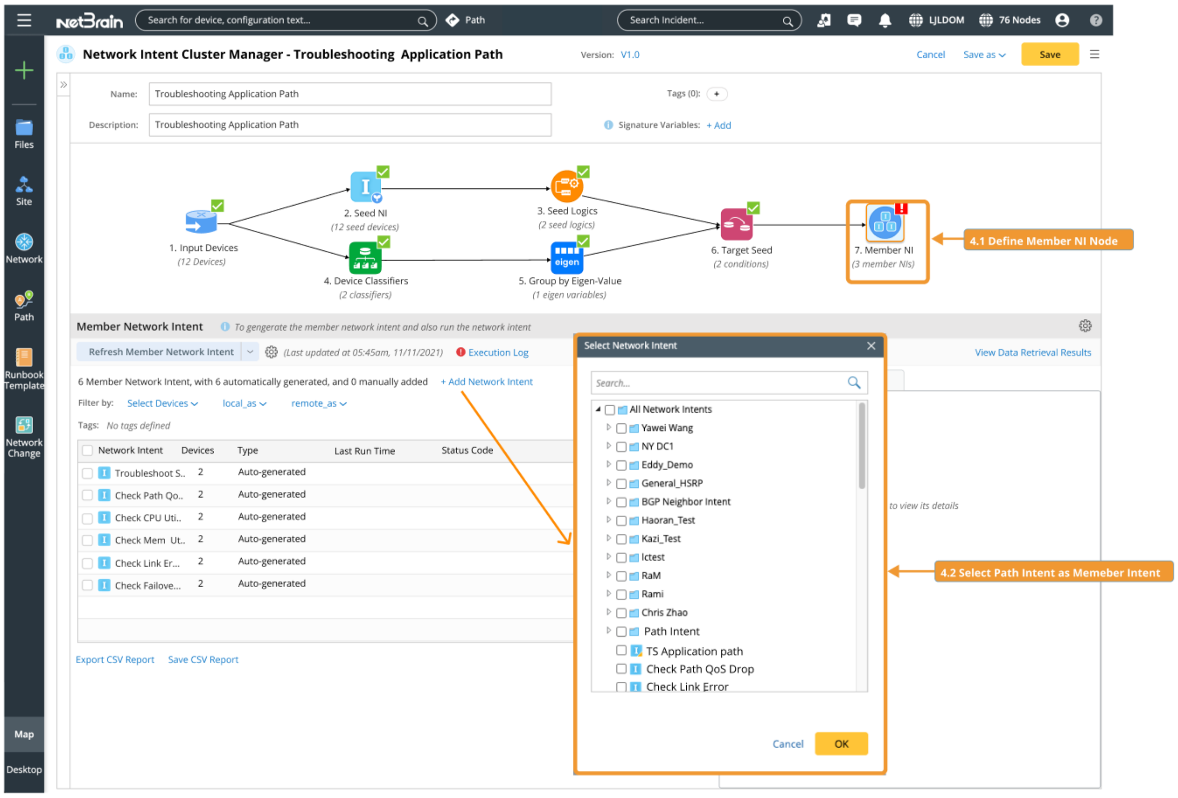
Same as step 1 of flow 1.

4. **Install Path-based Trigger Automation (When Network is Healthy)**

The Path NI can be triggered by TAF or Adaptive Monitoring. To install the Path NI for triggered automation, users need to:

1) Tag the Path NI with the path source/destination IP or DNS name, application name, and path name.

2) Add all Path NIs into a NIC as static member NIs.



Search for device, configuration text...
Path
Search Incident...
LJLDOM
76 Nodes
Files
Site
Network
Path
Runbook Template
Network Change
Map
Desktop
Network Intent Cluster Manager - Troubleshooting Application Path
Version: V1.0
Cancel
Save as
Save
1. Input Devices (12 Devices)
2. Seed NI (12 seed devices)
3. Seed Logics (2 seed logics)
4. Device Classifiers (2 classifiers)
5. Group by Eigen-Value (1 eigen variables)
6. Target Seed (2 conditions)
7. Member NI (3 member NIs)
4.1 Define Member NI Node
4.2 Select Path Intent as Memeber Intent
Refresh Member Network Intent
Execution Log
View Data Retrieval Results
Export CSV Report
Save CSV Report
Select Network Intent
Search...
Cancel
OK

Name: Troubleshooting Application Path

Description: Troubleshooting Application Path

Tags (0):

Signature Variables: + Add

Member Network Intent — To gengerate the member network intent and also run the network intent

(Last updated at 05:45am, 11/11/2021)

6 Member Network Intent, with 6 automatically generated, and 0 manually added + Add Network Intent

Filter by: Select Devices local_as remote_as

Tags: No tags defined

Network Intent	Devices	Type	Last Run Time	Status Code
Troubleshoot S..	2	Auto-generated		
Check Path Qo..	2	Auto-generated		
Check CPU Uti..	2	Auto-generated		
Check Mem Ut..	2	Auto-generated		
Check Link Er...	2	Auto-generated		
Check Failove...	2	Auto-generated		

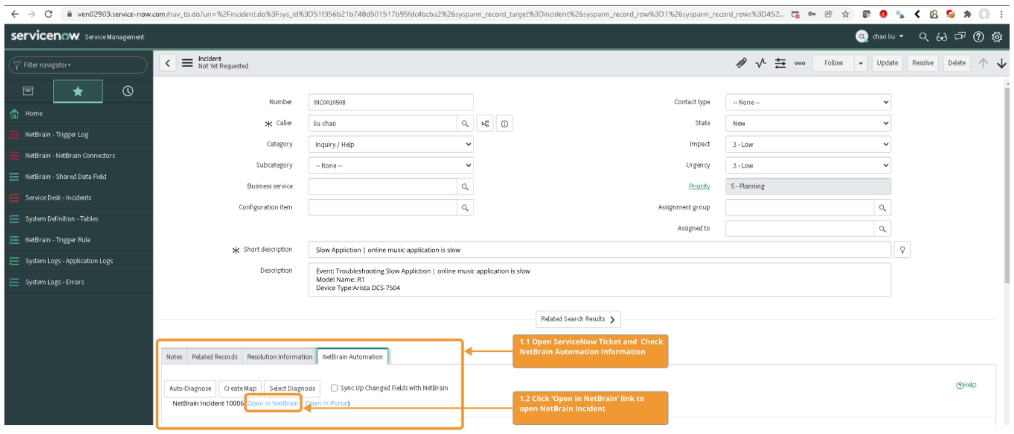
to view its details

- All Network Intents
  - Yawei Wang
  - NY DC1
  - Eddy_Demo
  - General_HSRP
  - BGP Neighbor Intent
  - Haoran_Test
  - Kazi_Test
  - lctest
  - RaM
  - Rami
  - Chris Zhao
  - Path Intent
    - TS Application path
    - Check Path QoS Drop
    - Check Link Error

3) Define trigger diagnosis using the NIC and the NIC filter condition with path NI tags (see section 2.2.3 for detail).

5. Open NetBrain Incident from ServiceNow Ticket (During troubleshooting)

A NetBrain incident can be created automatically for A ServiceNow ticket based on the TAF definition. NOC Engineers can open the ServiceNow app, check the ticket, and find the related link to the NetBrain Incident.


servicenow Service Management
Filter navigator
Home
NetBrain - Trigger Log
NetBrain - NetBrain Connectors
NetBrain - Shared Data Field
Service Desk - Incidents
System Definition - Tables
NetBrain - Trigger Rule
System Logs - Application Logs
System Logs - Errors
Follow
Update
Resolve
Delete
Related Search Results
Notes
Related Records
Resolution Information
NetBrain Automation
Auto-Diagnose
Create Map
Select Diagnosis
Sync Up Changed Fields with NetBrain
Open in NetBrain
Open in Portal
Help
1.1 Open ServiceNow Ticket and Check NetBrain Automation Information
1.2 Click 'Open in NetBrain' link to open NetBrain Incident

Incident
Not Yet Requested

Number	INC0010508	Contact type	-- None --
Caller	liu chao	State	New
Category	Inquiry / Help	Impact	3 - Low
Subcategory	-- None --	Urgency	3 - Low
Business service		Priority	5 - Planning
Configuration item		Assignment group	
		Assigned to	

Short description: Slow Appliction | online music application is slow

Description:
Event: Troubleshooting Slow Appliction | online music application is slow
Model Name: R1
Device Type:Arista DCS-7504

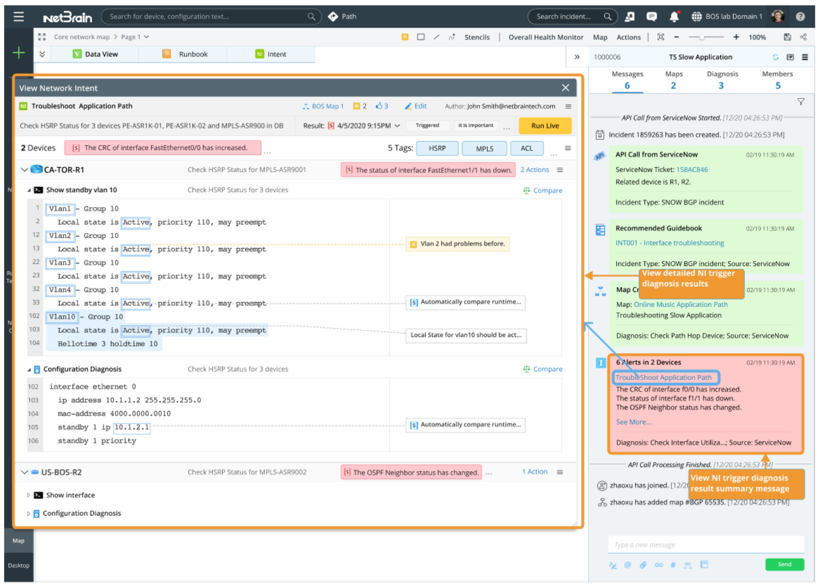
NetBrain Incident 10006 (Open in NetBrain Open in Portal)

6. Find documented Path and Review Path Documents (During troubleshooting)

A message containing a hyperlink to the path map will be displayed in the incident pane after the NI is triggered if a path map is saved as a NI reference map. Users can open the Path map, locate the Path from the map and view the path description, path note, path reference map, and Path logic.

7. Review Trigger Diagnosis Result (During troubleshooting)

Users can review the Path NI Definition to understand the trigger diagnosis result.


NetBrain
Search for device, configuration text...
Path
Search incident...
BOS lab Domain 1
Core network map > Page 1
Stencils
Overall Health Monitor
Map
Actions
100%
Data View
Runbook
Intent
View Network Intent
BOS Map 1
Edit
Run Live
Triggered
it is important
Compare
Vlan 2 had problems before.
Automatically compare runtime...
Local State for vlan10 should be act...
Map
Desktop
Type a new message
Send
View detailed NI trigger diagnosis results
View NI trigger diagnosis result summary message

Troubleshoot Application Path — Author: John Smith@netbraintech.com

Check HSRP Status for 3 devices PE-ASR1K-01, PE-ASR1K-02 and MPLS-ASR900 in DB — Result: [S] 4/5/2020 9:15PM

2 Devices — [S] The CRC of interface FastEthernet0/0 has increased. — 5 Tags: HSRP, MPLS, ACL

CA-TOR-R1 — Check HSRP Status for MPLS-ASR9001 — [S] The status of interface FastEthernet1/1 has down. — 2 Actions

Show standby vlan 10 — Check HSRP Status for 3 devices

```
Vlan1 - Group 10
  Local state is Active, priority 110, may preempt
Vlan2 - Group 10
  Local state is Active, priority 110, may preempt
Vlan3 - Group 10
  Local state is Active, priority 110, may preempt
Vlan4 - Group 10
  Local state is Active, priority 110, may preempt
Vlan10 - Group 10
  Local state is Active, priority 110, may preempt
  Hellotime 3 holdtime 10
```

Configuration Diagnosis — Check HSRP Status for 3 devices

```
interface ethernet 0
  ip address 10.1.1.2 255.255.255.0
  mac-address 4000.0000.0010
  standby 1 ip 10.1.2.1
  standby 1 priority
```

US-BOS-R2 — Check HSRP Status for MPLS-ASR9002 — [S] The OSPF Neighbor status has changed. — 1 Action

Show interface

Configuration Diagnosis

1000006 TS Slow Application

Messages	Maps	Diagnosis	Members
6	2	3	5

API Call from ServiceNow Started. [12/20 04:26:53 PM]

Incident 1859263 has been created. [12/20 04:26:53 PM]

API Call from ServiceNow 02/19 11:30:19 AM
ServiceNow Ticket: 158AC846
Related device is R1, R2.
Incident Type: SNOW BGP incident

Recommended Guidebook 02/19 11:30:19 AM
INT001 - Interface troubleshooting
Incident Type: SNOW BGP incident; Source: ServiceNow

Map Cr 02/19 11:30:19 AM
Map: Online Music Application Path
Troubleshooting Slow Application
Diagnosis: Check Path Hop Device; Source: ServiceNow

6 Alerts in 2 Devices 02/19 11:30:19 AM
TroubleShoot Application Path
The CRC of interface f0/0 has increased.
The status of interface f1/1 has down.
The OSPF Neighbor status has changed.
See More...
Diagnosis: Check Interface Utiliza...; Source: ServiceNow

API Call Processing Finished. [12/20 04:26:53 PM]

zhaoxu has joined. [12/2

zhaoxu has added map #BGP 65535. [12/20 04:26:53 PM]

8. **Drill-down analysis - Calculate live Path and Compare with Cached Path (During troubleshooting)**
   Same as step 5 of flow 1.

9. **Drill-down analysis - Execute Path intent (During troubleshooting)**
   Same with step 6 of flow 1.

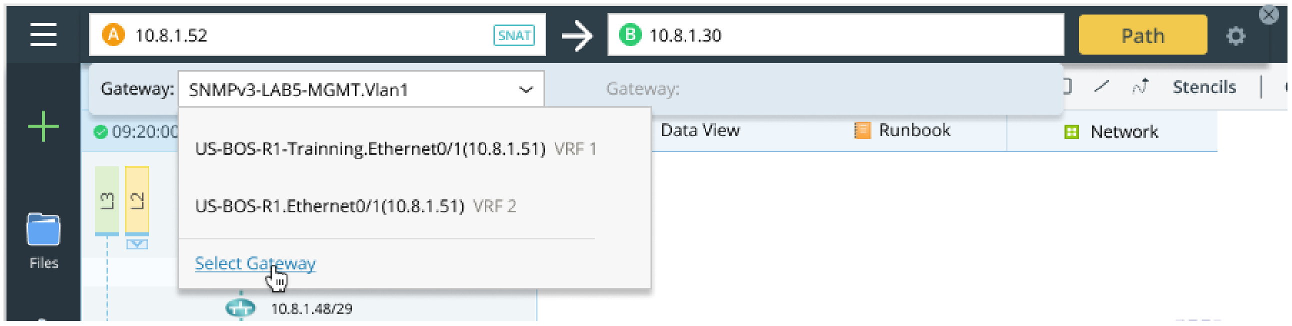
2.5.3 Main Enhancements

2.5.3.1 Discover Path

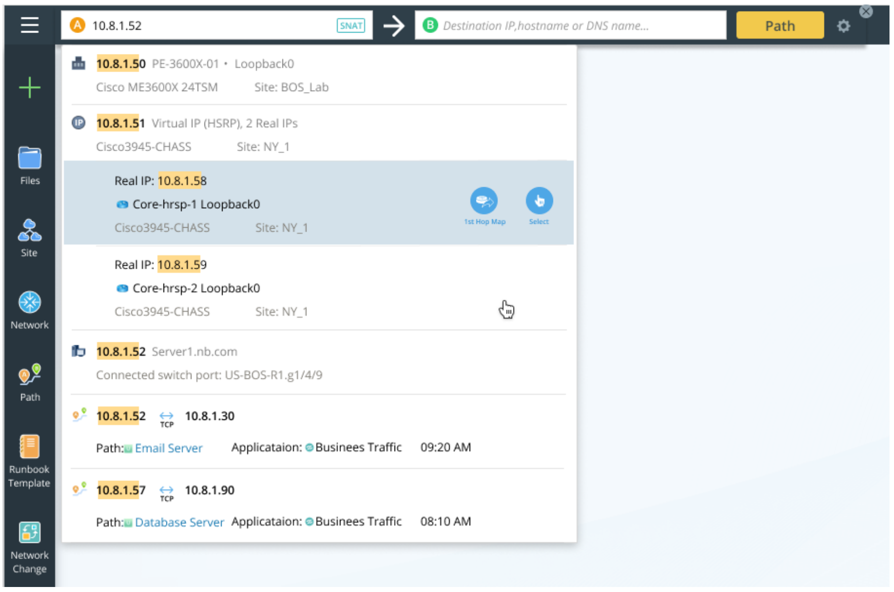
The user may have difficulty selecting the gateway (the first hop) while defining a path due to the duplicate IP, HSRP, HA group, etc. A typical scenario is: different end systems use the same IP address. Also, there is a limitation in previous versions: the system's list of gateways automatically created may not include the one the user wants to select.

In v10.1, users can select any device's interface with IP as the gateway for the end system (However, only the current device's interface with configured IP can be chosen as the gateway for a network device) as the following circumstances:

- The gateway auto-selected by tne system is not the one user wants.
- The user does not manage the gateway device.


A 10.8.1.52
SNAT
B 10.8.1.30
Path
Gateway: SNMPv3-LAB5-MGMT.Vlan1
Gateway:
Stencils
09:20:00
Data View
Runbook
Network
L3
L2
Files
US-BOS-R1-Trainning.Ethernet0/1(10.8.1.51) VRF 1
US-BOS-R1.Ethernet0/1(10.8.1.51) VRF 2
Select Gateway
10.8.1.48/29

The IntelliSense dialog while a user enters the IP addresses at Points A and B is improved to reduce the difficulty of the first hop choice. The following is an example when the user enters an IP address:


10.8.1.52 SNAT → Destination IP,hostname or DNS name... Path

10.8.1.50 PE-3600X-01 • Loopback0
Cisco ME3600X 24TSM Site: BOS_Lab

10.8.1.51 Virtual IP (HSRP), 2 Real IPs
Cisco3945-CHASS Site: NY_1

Real IP: 10.8.1.58
Core-hrsp-1 Loopback0
Cisco3945-CHASS Site: NY_1

Real IP: 10.8.1.59
Core-hrsp-2 Loopback0
Cisco3945-CHASS Site: NY_1

10.8.1.52 Server1.nb.com
Connected switch port: US-BOS-R1.g1/4/9

10.8.1.52 TCP 10.8.1.30
Path: Email Server Applicataion: Businees Traffic 09:20 AM

10.8.1.57 TCP 10.8.1.90
Path: Database Server Applicataion: Businees Traffic 08:10 AM

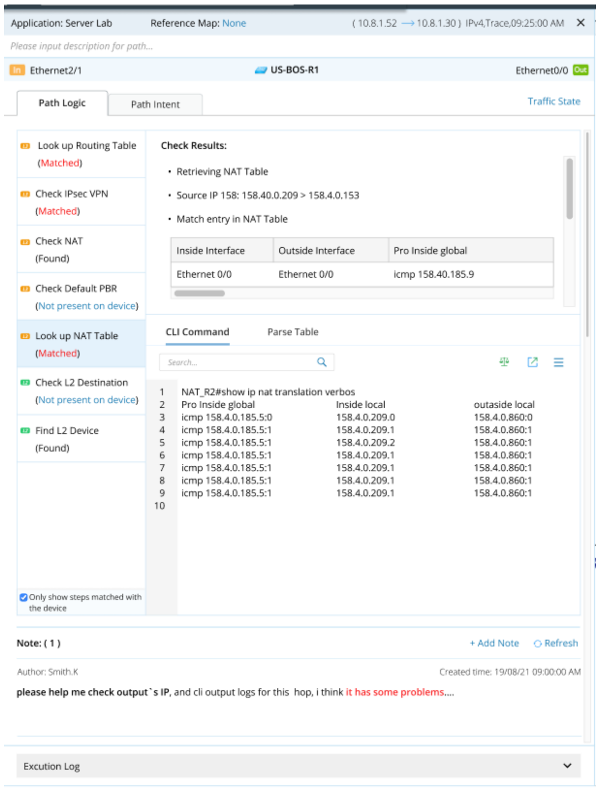
2.5.3.2 Document the Path

After mapping the application path while the network is working properly, users can document the path as follow:

- **Add path description and note**

The path description and note may describe the design and the possible issues a user wants to share with others.

In the Path Logic panel, the raw data used in each step of path discovery is shown. In addition, the user can add notes to describe their findings regarding each step of the path calculation.



Application: Server Lab   Reference Map: None   (10.8.1.52 ⟶ 10.8.1.30) IPv4,Trace,09:25:00 AM

Please input description for path...

In Ethernet2/1   US-BOS-R1   Ethernet0/0 Out

Path Logic | Path Intent | Traffic State

- Look up Routing Table (Matched)
- Check IPsec VPN (Matched)
- Check NAT (Found)
- Check Default PBR (Not present on device)
- Look up NAT Table (Matched)
- Check L2 Destination (Not present on device)
- Find L2 Device (Found)

☑ Only show steps matched with the device

Check Results:

- Retrieving NAT Table
- Source IP 158: 158.40.0.209 > 158.4.0.153
- Match entry in NAT Table

Inside Interface	Outside Interface	Pro Inside global
Ethernet 0/0	Ethernet 0/0	icmp 158.40.185.9

CLI Command | Parse Table

```
NAT_R2#show ip nat translation verbos
Pro Inside global           Inside local           outaside local
icmp 158.4.0.185.5:0        158.4.0.209.0          158.4.0.860:0
icmp 158.4.0.185.5:1        158.4.0.209.1          158.4.0.860:1
icmp 158.4.0.185.5:1        158.4.0.209.2          158.4.0.860:1
icmp 158.4.0.185.5:1        158.4.0.209.1          158.4.0.860:1
icmp 158.4.0.185.5:1        158.4.0.209.1          158.4.0.860:1
icmp 158.4.0.185.5:1        158.4.0.209.1          158.4.0.860:1
icmp 158.4.0.185.5:1        158.4.0.209.1          158.4.0.860:1
```

Note: (1)   + Add Note   Refresh

Author: Smith.K   Created time: 19/08/21 09:00:00 AM

please help me check output`s IP, and cli output logs for this hop, i think it has some problems....

Excution Log

- **Define and View Reference Map**

in the path details pane, a user can define the path reference map, in which he can draw the golden baseline path and add device notes for the key configurations.

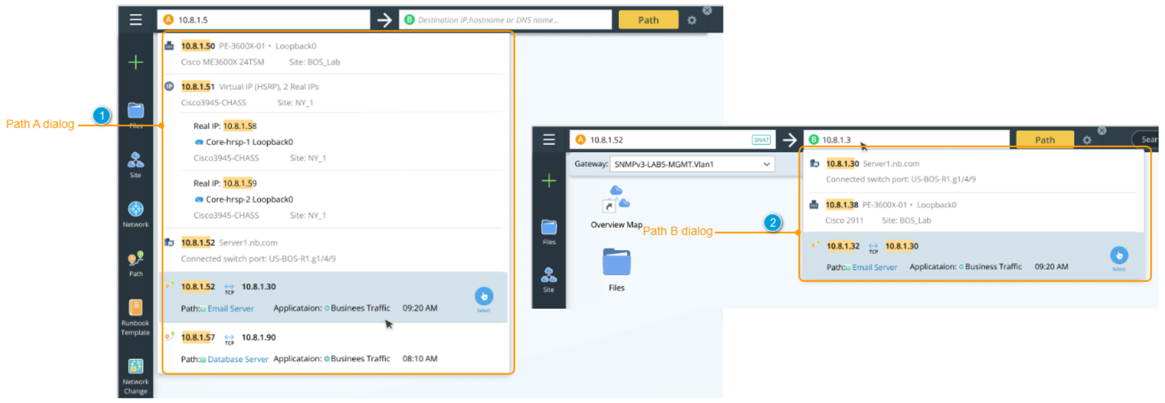
2.5.3.3 Define Path NI for Path Diagnose

The path discovery logic is based on network technologies such as routing, NAT, and ALC, which can be documented as an NI. Moreover, the user can define Configuration/CLI command diagnosis on each path hop. From the Path detail pane, a user can create a path NI. All path devices will be added to the NI edit page, each having a section to define the configuration or CLI command diagnosis.

A user can also select an existing NI to add it as the path NI. Then, he can sync this NI with the current path devices and set the current path map as the reference map.

2.5.3.4 Access the Saved Path via A/B Dialog

When a user enters the source or destination in the A/B dialog, the saved Path with the same source or destination will be displayed on the Intellisense dialog. The user can view the saved path results and open its path map.



A 10.8.1.5
B Destination IP,hostname or DNS name...

10.8.1.50 PE-3600X-01 • Loopback0
Cisco ME3600X 24TSM Site: BOS_Lab

10.8.1.51 Virtual IP (HSRP), 2 Real IPs
Cisco3945-CHASS Site: NY_1

Real IP: 10.8.1.58
Core-hrsp-1 Loopback0
Cisco3945-CHASS Site: NY_1

Real IP: 10.8.1.59
Core-hrsp-2 Loopback0
Cisco3945-CHASS Site: NY_1

10.8.1.52 Server1.nb.com
Connected switch port: US-BOS-R1.g1/4/9

10.8.1.52 TCP 10.8.1.30
Path: Email Server Applicataion: Businees Traffic 09:20 AM

10.8.1.57 TCP 10.8.1.90
Path: Database Server Applicataion: Businees Traffic 08:10 AM

A 10.8.1.52 SNAT
B 10.8.1.3
Gateway: SNMPv3-LAB5-MGMT.Vlan1

10.8.1.30 Server1.nb.com
Connected switch port: US-BOS-R1.g1/4/9

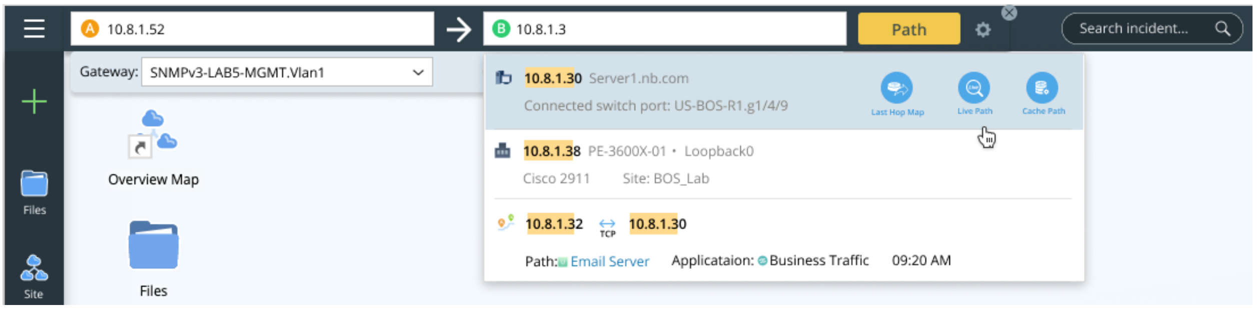
10.8.1.38 PE-3600X-01 • Loopback0
Cisco 2911 Site: BOS_Lab

10.8.1.32 TCP 10.8.1.30
Path: Email Server Applicataion: Business Traffic 09:20 AM

2.5.3.5 Discover the Live Path and Compare with the History

V10.1 made the following improvements for a user to discover the live path and compare it with the saved path:

- Provide an entry for the Live Path and Cached Path in the Intellisense B dialog so that a user can discover the Live or Cached Path without modifying the option.



A 10.8.1.52
B 10.8.1.3
Gateway: SNMPv3-LAB5-MGMT.Vlan1

10.8.1.30 Server1.nb.com
Connected switch port: US-BOS-R1.g1/4/9

10.8.1.38 PE-3600X-01 • Loopback0
Cisco 2911 Site: BOS_Lab

10.8.1.32 TCP 10.8.1.30
Path: Email Server Applicataion: Business Traffic 09:20 AM

- Browse historical paths

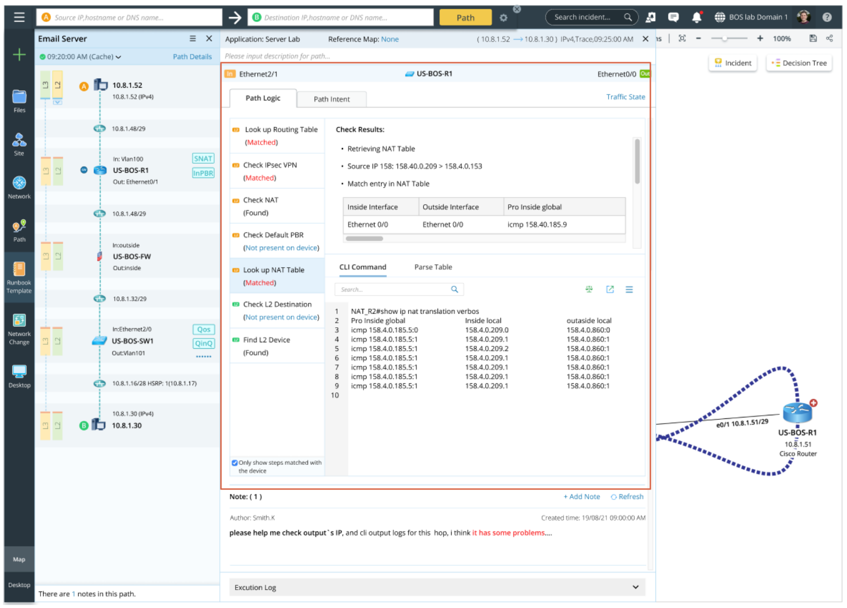
In the path result pane, a user can browse all historical paths, check whether the path is golden, and set/remove a path as golden. The current path will be auto-saved and listed as the historical path later.

- Run Path NI to check the Diagnosis result

The user can execute the path NI and view its results in the new Path NI pane.

2.5.3.6 Improve Usability of UI Result and Detail Pane

- Display the check results and the raw data (raw CLI data, NCT, system table, or configurations) in each hop.



Application: Server Lab   Reference Map: None   (10.8.1.52 → 10.8.1.30) IPv4,Trace,09:25:00 AM

Please input description for path...

In Ethernet2/1   US-BOS-R1   Ethernet0/0 Out

Look up Routing Table (Matched)
Check IPsec VPN (Matched)
Check NAT (Found)
Check Default PBR (Not present on device)
Look up NAT Table (Matched)
Check L2 Destination (Not present on device)
Find L2 Device (Found)
Only show steps matched with the device

Check Results:

- Retrieving NAT Table
- Source IP 158: 158.40.0.209 > 158.4.0.153
- Match entry in NAT Table

Inside Interface	Outside Interface	Pro Inside global
Ethernet 0/0	Ethernet 0/0	icmp 158.40.185.9

```
NAT_R2#show ip nat translation verbos
Pro Inside global          Inside local        outaside local
icmp 158.4.0.185.5:0       158.4.0.209.0       158.4.0.860:0
icmp 158.4.0.185.5:1       158.4.0.209.1       158.4.0.860:1
icmp 158.4.0.185.5:1       158.4.0.209.2       158.4.0.860:1
icmp 158.4.0.185.5:1       158.4.0.209.1       158.4.0.860:1
icmp 158.4.0.185.5:1       158.4.0.209.1       158.4.0.860:1
icmp 158.4.0.185.5:1       158.4.0.209.1       158.4.0.860:1
icmp 158.4.0.185.5:1       158.4.0.209.1       158.4.0.860:1
```

Note: (1)

Author: Smith.K   Created time: 19/08/21 09:00:00 AM

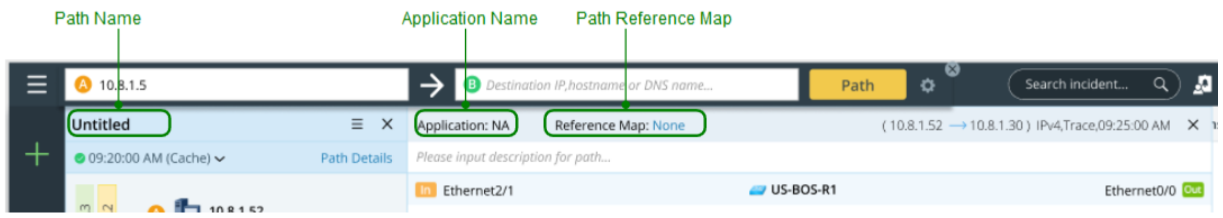
please help me check output`s IP, and cli output logs for this hop, i think it has some problems....

There are 1 notes in this path.

- Add the tag on each path hop for the network technology checked by the path discovery logic.

- Display the path name, application name, and path reference map.

  The user can tell whether a path is saved by the path name being untitled or not. The user can also set the reference map.



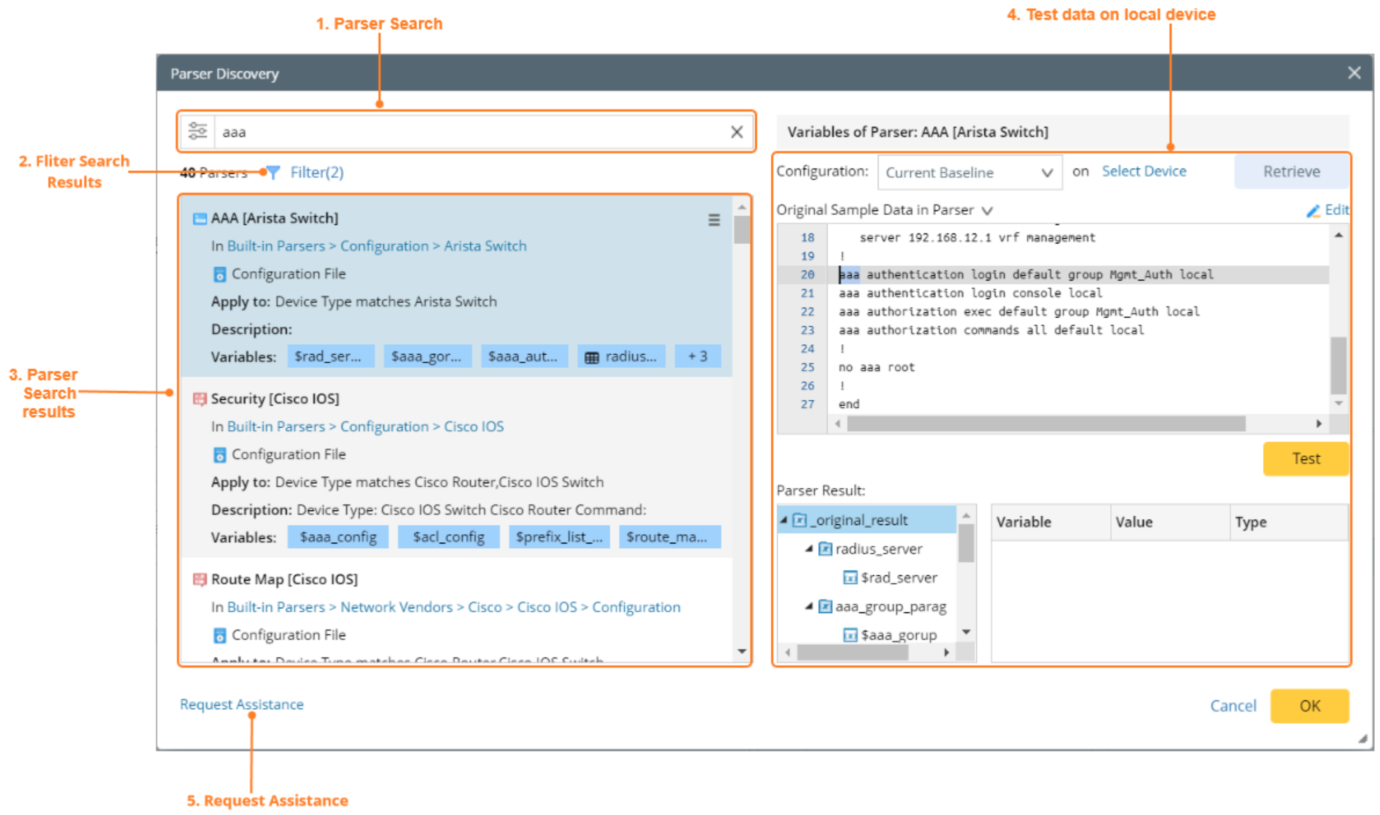
- Only block the Path Result pane instead of the whole UI to allow the user to access other operations during the path calculation.

2.6 Parser Discovery

The Parsers are the building blocks of NetBrain automation such as NI, DVT, and Qapp. A new feature of v10.1, Parser Discovery, allows the user to quickly find a parser he needs from the parser library and test the Parser with the local data. If the user cannot find the Parser, he can create a Visual Parser or submit a ticket to request NetBrain to create one.

The Parser Discovery has the following key functions:

- Search Parser by parser name, description, CLI command, variable name, and sample text.
- Filter Parser to narrow down the search results with device and parser types.
- Test the Parser on a local device.
- Can request assistance from NetBrain if the Parser is not found.



Parser Discovery

aaa

40 Parsers Filter(2)

AAA [Arista Switch]
In Built-in Parsers > Configuration > Arista Switch
Configuration File
Apply to: Device Type matches Arista Switch
Description:
Variables: $rad_ser... $aaa_gor... $aaa_aut... radius... + 3

Security [Cisco IOS]
In Built-in Parsers > Configuration > Cisco IOS
Configuration File
Apply to: Device Type matches Cisco Router,Cisco IOS Switch
Description: Device Type: Cisco IOS Switch Cisco Router Command:
Variables: $aaa_config $acl_config $prefix_list_... $route_ma...

Route Map [Cisco IOS]
In Built-in Parsers > Network Vendors > Cisco > Cisco IOS > Configuration
Configuration File

Variables of Parser: AAA [Arista Switch]

Configuration: Current Baseline on Select Device Retrieve

Original Sample Data in Parser Edit

```
   server 192.168.12.1 vrf management
!
aaa authentication login default group Mgmt_Auth local
aaa authentication login console local
aaa authorization exec default group Mgmt_Auth local
aaa authorization commands all default local
!
no aaa root
!
end
```

Test

Parser Result:

_original_result
  radius_server
    $rad_server
  aaa_group_parag
    $aaa_gorup

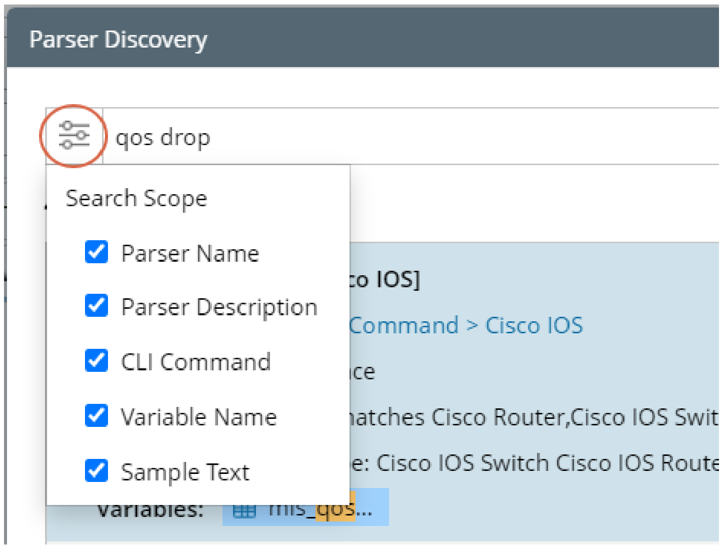
Variable	Value	Type

Request Assistance Cancel OK

2.6.1 Search a Parser

Parser Discovery is used whenever a user selects a Parser, allowing users to search parsers by Parser fields, filter the results by the device/Parser type, and do a quick test with the local data.

V10.1 improves the usability of searching parser. By default, users can search parsers by parser name, parser description, CLI command, variable name, and sample text:


Parser Discovery

qos drop

Search Scope

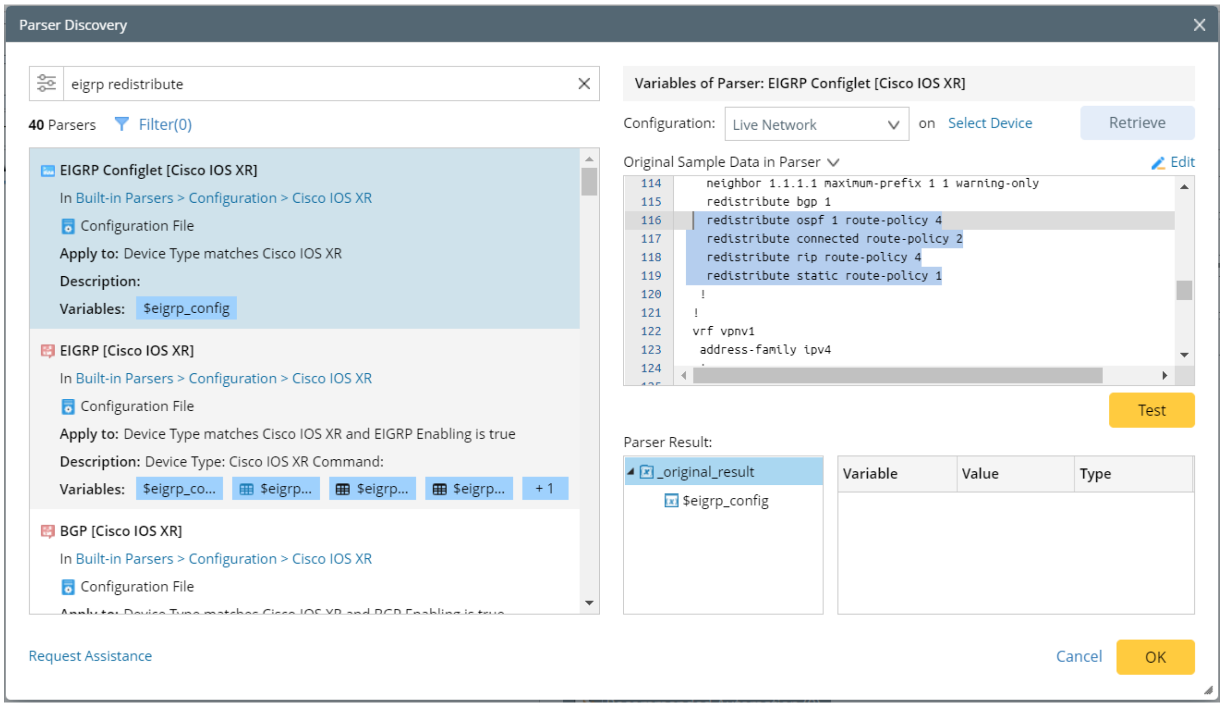
- ✓ Parser Name
- ✓ Parser Description
- ✓ CLI Command
- ✓ Variable Name
- ✓ Sample Text

co IOS]
Command > Cisco IOS
ce
atches Cisco Router,Cisco IOS Swit
e: Cisco IOS Switch Cisco IOS Route
Variables: mis_qos...

Users can specify the search scope to achieve a more precise search:

- **Search Configuration Parser**

If the user wants to search for a configuration parser, he can set Sample Data (Text) as the search scope. e.g., using keywords ***eigrp resitribute*** contained in its sample data.


Parser Discovery

eigrp redistribute

40 Parsers   Filter(0)

EIGRP Configlet [Cisco IOS XR]
In Built-in Parsers > Configuration > Cisco IOS XR
Configuration File
Apply to: Device Type matches Cisco IOS XR
Description:
Variables: $eigrp_config

EIGRP [Cisco IOS XR]
In Built-in Parsers > Configuration > Cisco IOS XR
Configuration File
Apply to: Device Type matches Cisco IOS XR and EIGRP Enabling is true
Description: Device Type: Cisco IOS XR Command:
Variables: $eigrp_co... $eigrp... $eigrp... $eigrp... +1

BGP [Cisco IOS XR]
In Built-in Parsers > Configuration > Cisco IOS XR
Configuration File

Variables of Parser: EIGRP Configlet [Cisco IOS XR]

Configuration: Live Network on Select Device   Retrieve

Original Sample Data in Parser   Edit

```
   neighbor 1.1.1.1 maximum-prefix 1 1 warning-only
   redistribute bgp 1
   redistribute ospf 1 route-policy 4
   redistribute connected route-policy 2
   redistribute rip route-policy 4
   redistribute static route-policy 1
  !
 !
vrf vpnv1
 address-family ipv4
```

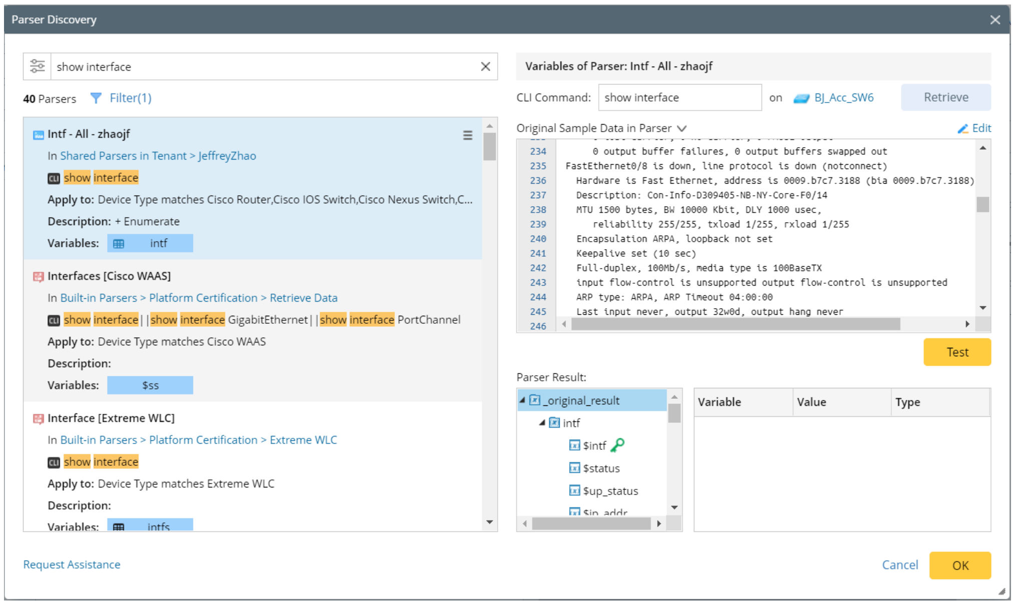
Parser Result:

_original_result
  $eigrp_config

Variable	Value	Type

- Search CLI Parsers

If the user wants to search for a CLI command parser, he can set the CLI command as search scope. e.g., using keyword ***show interface*** contained in its CLI command.



Parser Discovery

show interface

40 Parsers Filter(1)

Intf - All - zhaojf
In Shared Parsers in Tenant > JeffreyZhao
show interface
Apply to: Device Type matches Cisco Router,Cisco IOS Switch,Cisco Nexus Switch,C...
Description: + Enumerate
Variables: intf

Interfaces [Cisco WAAS]
In Built-in Parsers > Platform Certification > Retrieve Data
show interface| |show interface GigabitEthernet| |show interface PortChannel
Apply to: Device Type matches Cisco WAAS
Description:
Variables: $ss

Interface [Extreme WLC]
In Built-in Parsers > Platform Certification > Extreme WLC
show interface
Apply to: Device Type matches Extreme WLC
Description:
Variables: intfs

Variables of Parser: Intf - All - zhaojf

CLI Command: show interface on BJ_Acc_SW6

Original Sample Data in Parser

```
    0 output buffer failures, 0 output buffers swapped out
FastEthernet0/8 is down, line protocol is down (notconnect)
  Hardware is Fast Ethernet, address is 0009.b7c7.3188 (bia 0009.b7c7.3188)
  Description: Con-Info-D309405-NB-NY-Core-F0/14
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 100Mb/s, media type is 100BaseTX
  input flow-control is unsupported output flow-control is unsupported
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output 32w0d, output hang never
```

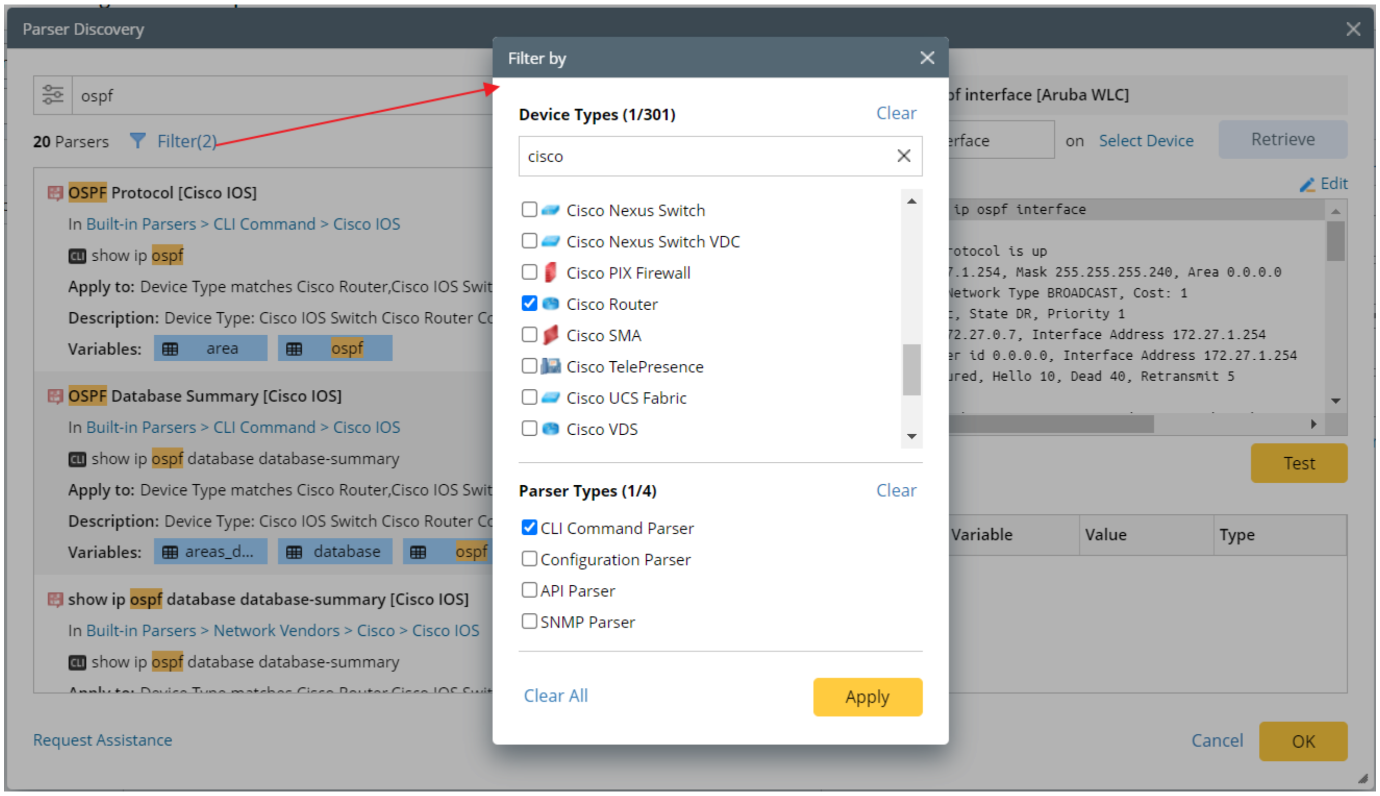
Parser Result:

- _original_result
  - intf
    - $intf
    - $status
    - $up_status
    - $ip_addr

Variable	Value	Type

2.6.2 Filter Parsers by Device Types and Parser Types

Through Filter, the user can quickly narrow down the search results by device type and parser type, e.g., filtering matched CLI parsers only applied to Cisco Router.


Parser Discovery

ospf

20 Parsers Filter(2)

OSPF Protocol [Cisco IOS]

In Built-in Parsers > CLI Command > Cisco IOS

show ip ospf

Apply to: Device Type matches Cisco Router,Cisco IOS Swit

Description: Device Type: Cisco IOS Switch Cisco Router C

Variables: area ospf

OSPF Database Summary [Cisco IOS]

In Built-in Parsers > CLI Command > Cisco IOS

show ip ospf database database-summary

Apply to: Device Type matches Cisco Router,Cisco IOS Swit

Description: Device Type: Cisco IOS Switch Cisco Router C

Variables: areas_d... database ospf

show ip ospf database database-summary [Cisco IOS]

In Built-in Parsers > Network Vendors > Cisco > Cisco IOS

show ip ospf database database-summary

Filter by

Device Types (1/301) Clear

cisco

- [ ] Cisco Nexus Switch
- [ ] Cisco Nexus Switch VDC
- [ ] Cisco PIX Firewall
- [x] Cisco Router
- [ ] Cisco SMA
- [ ] Cisco TelePresence
- [ ] Cisco UCS Fabric
- [ ] Cisco VDS

Parser Types (1/4) Clear

- [x] CLI Command Parser
- [ ] Configuration Parser
- [ ] API Parser
- [ ] SNMP Parser

Clear All Apply

f interface [Aruba WLC]

rface on Select Device Retrieve

Edit

```
ip ospf interface

rotocol is up
.1.254, Mask 255.255.255.240, Area 0.0.0.0
Network Type BROADCAST, Cost: 1
, State DR, Priority 1
72.27.0.7, Interface Address 172.27.1.254
r id 0.0.0.0, Interface Address 172.27.1.254
red, Hello 10, Dead 40, Retransmit 5
```

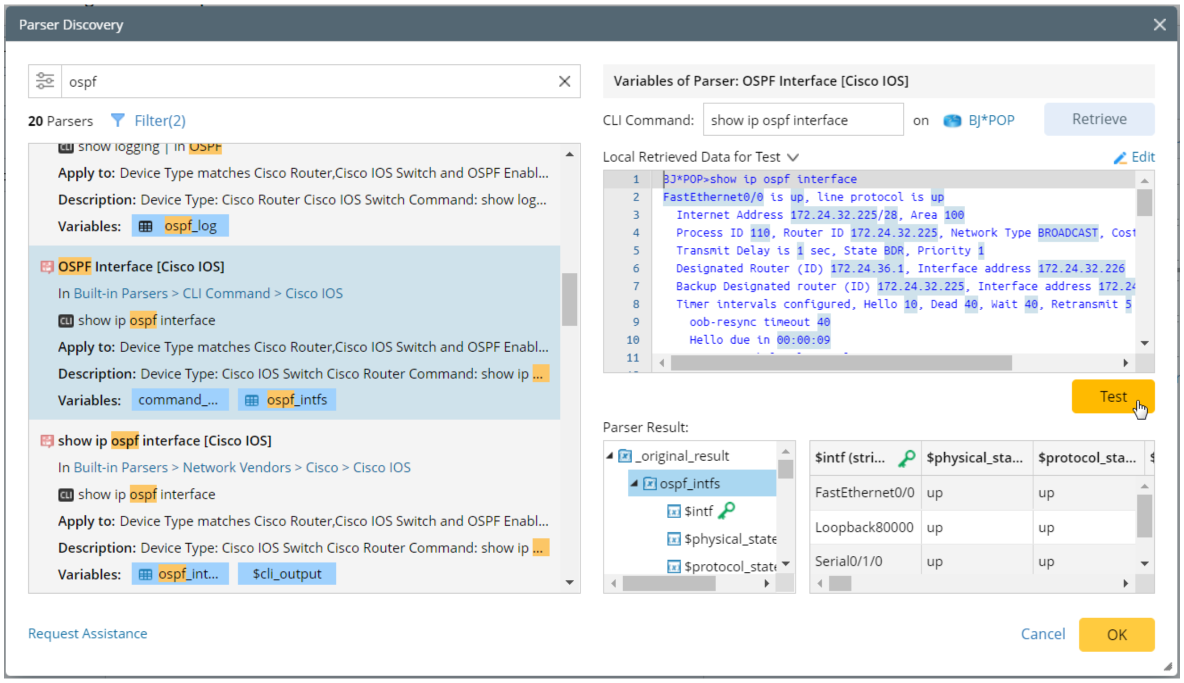
Test

Variable	Value	Type

Request Assistance Cancel OK

2.6.3 Test Parser with the local data

The user can select a device from the domain to retrieve the data, then test the Parser with the retrieved data to view if the Parser can parse variables correctly.



show logging | in OSPF
Apply to: Device Type matches Cisco Router,Cisco IOS Switch and OSPF Enabl...
Description: Device Type: Cisco Router Cisco IOS Switch Command: show log...
Variables: ospf_log

OSPF Interface [Cisco IOS]
In Built-in Parsers > CLI Command > Cisco IOS
show ip ospf interface
Apply to: Device Type matches Cisco Router,Cisco IOS Switch and OSPF Enabl...
Description: Device Type: Cisco IOS Switch Cisco Router Command: show ip ...
Variables: command_... ospf_intfs

show ip ospf interface [Cisco IOS]
In Built-in Parsers > Network Vendors > Cisco > Cisco IOS
show ip ospf interface
Apply to: Device Type matches Cisco Router,Cisco IOS Switch and OSPF Enabl...
Description: Device Type: Cisco IOS Switch Cisco Router Command: show ip ...
Variables: ospf_int... $cli_output

Variables of Parser: OSPF Interface [Cisco IOS]

CLI Command: show ip ospf interface on BJ*POP

Local Retrieved Data for Test

```
BJ*POP>show ip ospf interface
FastEthernet0/0 is up, line protocol is up
  Internet Address 172.24.32.225/28, Area 100
  Process ID 110, Router ID 172.24.32.225, Network Type BROADCAST, Cost
  Transmit Delay is 1 sec, State BDR, Priority 1
  Designated Router (ID) 172.24.36.1, Interface address 172.24.32.226
  Backup Designated router (ID) 172.24.32.225, Interface address 172.24
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:09
```

Parser Result:

_original_result
  ospf_intfs
    $intf
    $physical_state
    $protocol_state

$intf (stri...	$physical_sta...	$protocol_sta...
FastEthernet0/0	up	up
Loopback80000	up	up
Serial0/1/0	up	up

2.6.4 Request Assistance from NetBrain

When the user cannot find the Parser, he can submit a ticket to NetBrain by clicking Request Assistance. Fill in the description and attach the CLI command output (the retrieved sample data) to the ticket. When the ticket is submitted, the NetBrain team will create the Parser and deliver it to the user by Knowledge Cloud (KC) or email.

2.7 Visual Parser Improvements

V10.1 expands the function of the visual parser to enable the visual parser to support SNMP data and adds the support of the collector parser group to solve the cases where the paragraph parser group cannot parse data.

The collector parser group can find all matching items based on the defined keywords, automatically group them according to different values of the keywords, and parse the required variables based on the grouping.

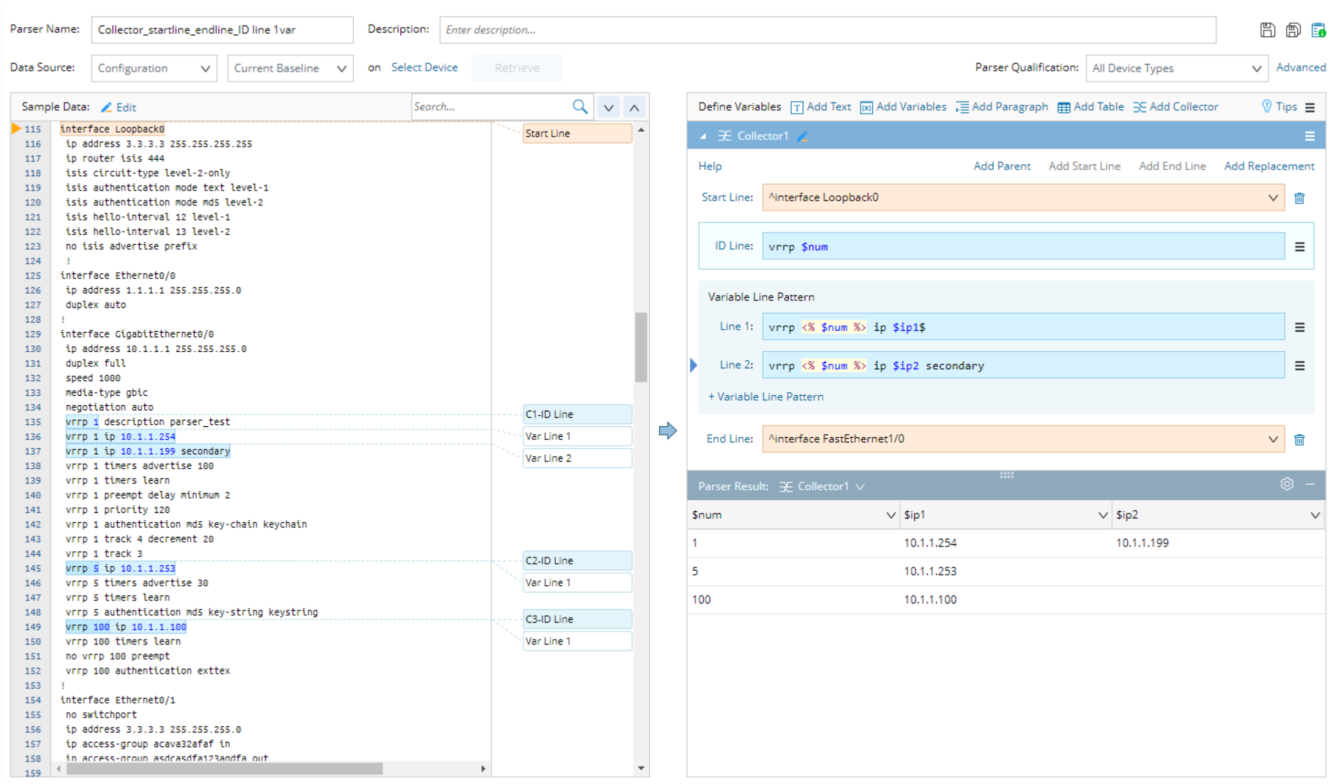
Collector parser group solves the following types of cases:

- The configuration scenario of VRRP/IS-IS: configurations recur in segments based on keywords, and all belong to a group of data.

- The configuration files in Juniper and some Firewalls are defined in JSON format such as { } and not easy to parse.

- Provide the capability to group during log file parsing.

The collector group definition is as follows:



```
interface Loopback0
 ip address 3.3.3.3 255.255.255.255
 ip router isis 444
 isis circuit-type level-2-only
 isis authentication mode text level-1
 isis authentication mode md5 level-2
 isis hello-interval 12 level-1
 isis hello-interval 13 level-2
 no isis advertise prefix
 !
interface Ethernet0/0
 ip address 1.1.1.1 255.255.255.0
 duplex auto
!
interface GigabitEthernet0/0
 ip address 10.1.1.1 255.255.255.0
 duplex full
 speed 1000
 media-type gbic
 negotiation auto
 vrrp 1 description parser_test
 vrrp 1 ip 10.1.1.254
 vrrp 1 ip 10.1.1.199 secondary
 vrrp 1 timers advertise 100
 vrrp 1 timers learn
 vrrp 1 preempt delay minimum 2
 vrrp 1 priority 120
 vrrp 1 authentication md5 key-chain keychain
 vrrp 1 track 4 decrement 20
 vrrp 1 track 3
 vrrp 5 ip 10.1.1.253
 vrrp 5 timers advertise 30
 vrrp 5 timers learn
 vrrp 5 authentication md5 key-string keystring
 vrrp 100 ip 10.1.1.100
 vrrp 100 timers learn
 no vrrp 100 preempt
 vrrp 100 authentication exttex
!
interface Ethernet0/1
 no switchport
 ip address 3.3.3.3 255.255.255.0
 ip access-group acava32afaf in
 ip access-group asdcasdfa123agdfa out
```

Start Line: ^interface Loopback0

ID Line: vrrp $num

Variable Line Pattern

Line 1: vrrp <% $num %> ip $ip1$

Line 2: vrrp <% $num %> ip $ip2 secondary

End Line: ^interface FastEthernet1/0

$num	$ip1	$ip2
1	10.1.1.254	10.1.1.199
5	10.1.1.253	
100	10.1.1.100	

ID Line: define where the ID variable(s) in the collector group is located. Only define ID variables, not other non-ID variables. All variables in the ID line should be ID variables and must be matched successfully. Then group the entire parser input text according to the actual value of the ID variable.

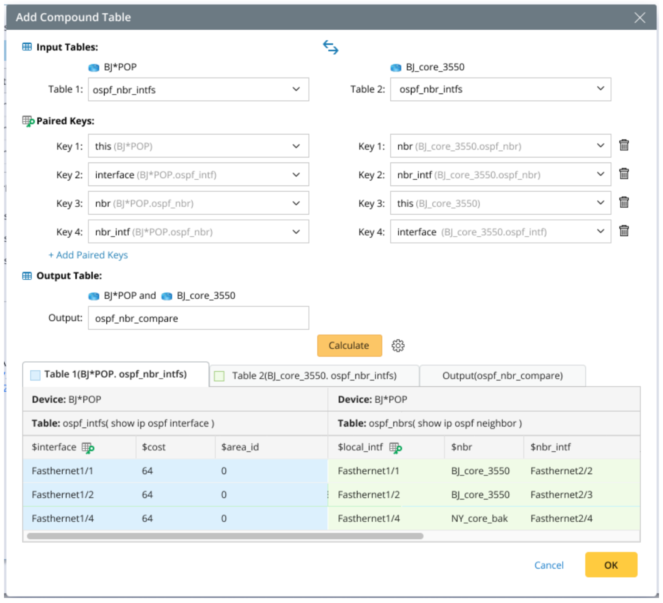
Variable Lines: If users want to parse specific variable values of different groups, they need to define a variable line that repeatedly refers to ID Variable to confirm which values are parsed. In the variable line of collector, a specific syntax is supported to refer to the variable defined in the ID Line, for example ***<% $var1 %>.*** Users can also define multiple variable lines in the collector.

2.8 Network Intent Improvements

V10.1 improved NI by enabling the table diagnosis between devices with no-code automation, enhancing the export and import functionality, and adding the test run with baseline data. These NI improvements provide a smoother workflow to define NI for customers with no code automation.

2.8.1 Add Compound Table

The **Add Compound Table** feature replaces the old **merge table** with more functions and a user-friendly design. The tables can be from the same devices or two different devices. For example, Table 1 is an OSPF neighbor interface table ***ospf_nbr_intfs*** from one device, and Table 2 is an OSPF neighbor interface table ***ospf_nbr_intfs*** from another OSPF neighbor device. The two tables can be merged using ***this, nbr, interface,*** and ***nbr_intf*** as the paired Key into an OSPF neighbor pair compound table.


Add Compound Table

Input Tables:

BJ*POP		BJ_core_3550	
Table 1:	ospf_nbr_intfs	Table 2:	ospf_nbr_intfs

Paired Keys:

Key 1:	this (BJ*POP)	Key 1:	nbr (BJ_core_3550.ospf_nbr)
Key 2:	interface (BJ*POP.ospf_intf)	Key 2:	nbr_intf (BJ_core_3550.ospf_nbr)
Key 3:	nbr (BJ*POP.ospf_nbr)	Key 3:	this (BJ_core_3550)
Key 4:	nbr_intf (BJ*POP.ospf_nbr)	Key 4:	interface (BJ_core_3550.ospf_intf)

+ Add Paired Keys

Output Table:

BJ*POP and BJ_core_3550

Output: ospf_nbr_compare

Calculate
Table 1(BJ*POP. ospf_nbr_intfs)
Table 2(BJ_core_3550. ospf_nbr_intfs)
Output(ospf_nbr_compare)

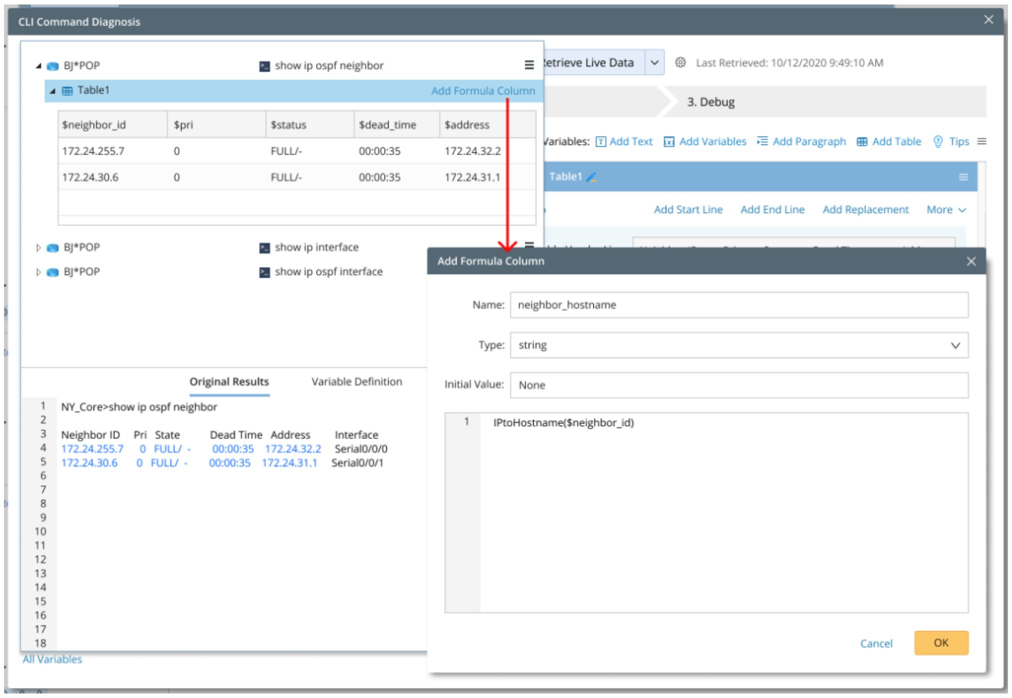
Device: BJ*POP			**Device:** BJ*POP		
Table: ospf_intfs(show ip ospf interface)			**Table:** ospf_nbrs(show ip ospf neighbor)		
$interface	$cost	$area_id	$local_intf	$nbr	$nbr_intf
Fasthernet1/1	64	0	Fasthernet1/1	BJ_core_3550	Fasthernet2/2
Fasthernet1/2	64	0	Fasthernet1/2	BJ_core_3550	Fasthernet2/3
Fasthernet1/4	64	0	Fasthernet1/4	NY_core_bak	Fasthernet2/4

This compound table contains information about the neighbor paired device, interface, OSPF cost, and area ID.

The output table can be adjusted in Settings to remove the columns or rows without matching key columns. Compound tables can be selected in NI Diagnosis as part of the Diagnosis Conditions.

2.8.2 Add Formula Column in Table

Users can add the Formula Column on the table variable from the parser and the compound table. Formula Column converts the variables into different formats or values in new columns. For example, convert the IP address to the corresponding device hostname, or add two variables via simple arithmetic expressions, ***$var1 + $var2***.



$neighbor_id	$pri	$status	$dead_time	$address
172.24.255.7	0	FULL/-	00:00:35	172.24.32.2
172.24.30.6	0	FULL/-	00:00:35	172.24.31.1

```
NY_Core>show ip ospf neighbor

Neighbor ID    Pri  State       Dead Time  Address       Interface
172.24.255.7    0   FULL/ -     00:00:35   172.24.32.2   Serial0/0/0
172.24.30.6     0   FULL/ -     00:00:35   172.24.31.1   Serial0/0/1
```

Add Formula Column

Name: neighbor_hostname

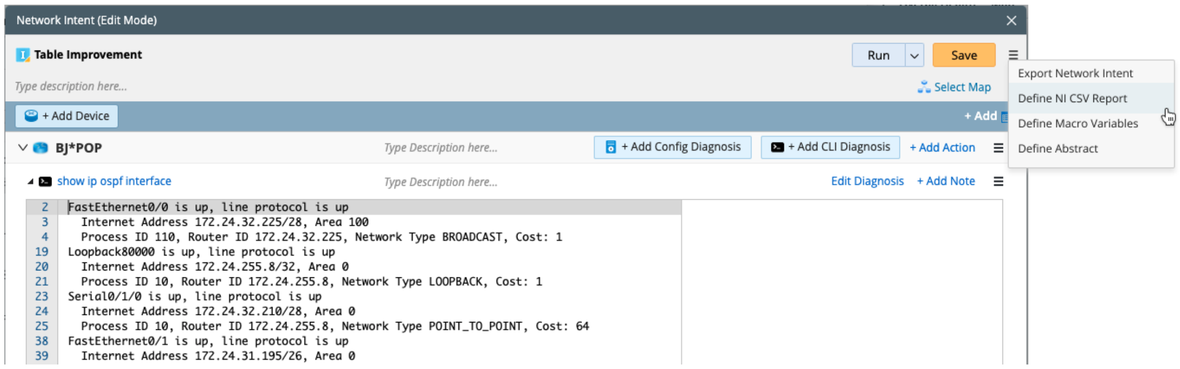
Type: string

Initial Value: None

```
IPtoHostname($neighbor_id)
```

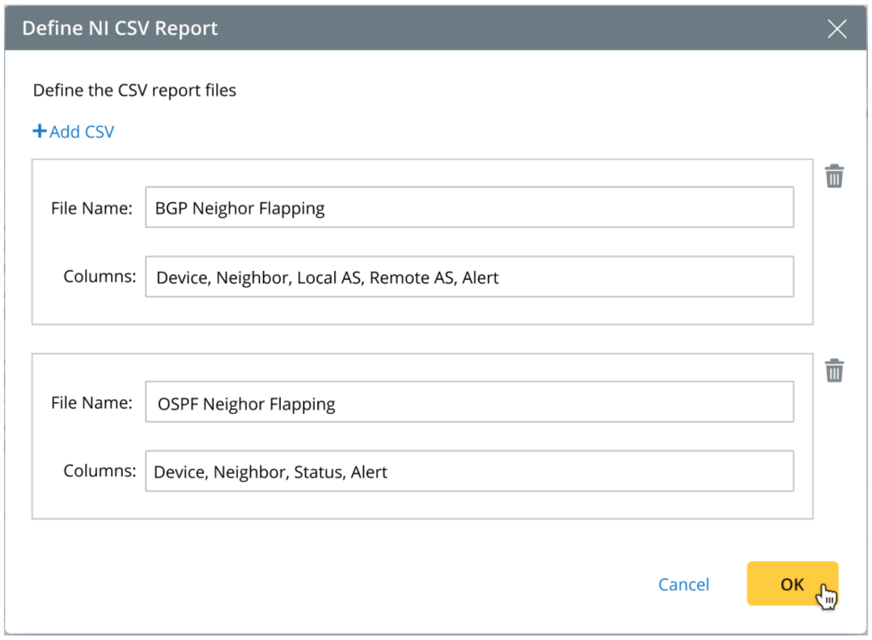
2.8.3 Export CSV Report

The network data parsed by the parser and the status code from the previous execution result can be organized into a CSV report and saved in the NetBrain file system or downloaded to a local computer.



```
FastEthernet0/0 is up, line protocol is up
  Internet Address 172.24.32.225/28, Area 100
  Process ID 110, Router ID 172.24.32.225, Network Type BROADCAST, Cost: 1
Loopback80000 is up, line protocol is up
  Internet Address 172.24.255.8/32, Area 0
  Process ID 10, Router ID 172.24.255.8, Network Type LOOPBACK, Cost: 1
Serial0/1/0 is up, line protocol is up
  Internet Address 172.24.32.210/28, Area 0
  Process ID 10, Router ID 172.24.255.8, Network Type POINT_TO_POINT, Cost: 64
FastEthernet0/1 is up, line protocol is up
  Internet Address 172.24.31.195/26, Area 0
```

Users can define the CSV file name and the column names in the CSV report. One NI can have multiple CSV reports.



File Name:	BGP Neighor Flapping
Columns:	Device, Neighbor, Local AS, Remote AS, Alert
File Name:	OSPF Neighor Flapping
Columns:	Device, Neighbor, Status, Alert

The users can select the data to be exported and map it to the CSV column. The following data can be exported:

- Built-in Data: built-in data from NI, such as ***$this_device*** (the device name for current device), Diagnosis Note, Device Status Code, Intent Status Code, device status code, and intent status code.

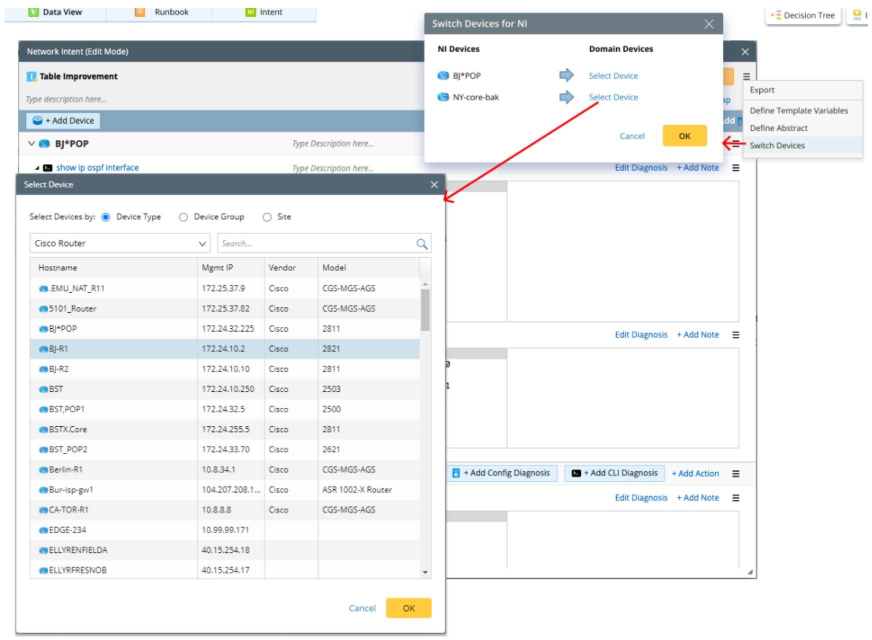
- Variables: If **Loop Table Rows** is checked, the table variables selected for the current diagnosis and all the single-valued variables for the current device can be selected and exported to the CSV report. Otherwise, only single-valued variables can be selected and exported for the current device.
- Macro Variable: All Macro Variables for the current device can be exported.

The generated CSV report can be viewed in the Network Intent View Mode after NI is executed. If NI is executed in a Runbook, the generated CSV report can be Exported or Saved by clicking on the CSV report name in the NI result pane.

The CSV reports of a NIC's member NIs can be merged into one report when exporting or saving the report.

2.8.4 Switch Devices in Device Section

The devices defined in a NI may not exist in the customer's network, and users can switch to the devices in their domain before using the sample NI. After the devices are switched, the device scope will be changed to the selected devices. The baseline data will be switched automatically to the baseline data of the selected devices. The devices used in diagnosis, compound variable, compound table, and Formula Column will also be changed automatically.



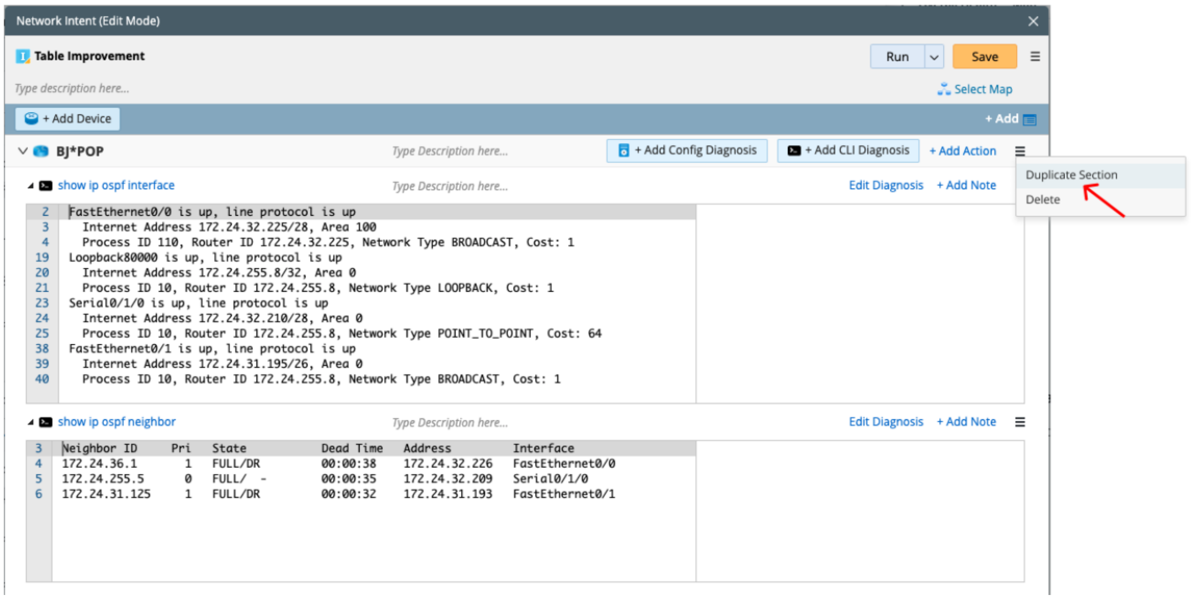
Select Devices by: [✓ Device Type] Device Group Site

Cisco Router

Hostname	Mgmt IP	Vendor	Model
.EMU_NAT_R11	172.25.37.9	Cisco	CGS-MGS-AGS
5101_Router	172.25.37.82	Cisco	CGS-MGS-AGS
BJ*POP	172.24.32.225	Cisco	2811
BJ-R1	172.24.10.2	Cisco	2821
BJ-R2	172.24.10.10	Cisco	2811
BST	172.24.10.250	Cisco	2503
BST,POP1	172.24.32.5	Cisco	2500
BSTX.Core	172.24.255.5	Cisco	2811
BST_POP2	172.24.33.70	Cisco	2621
Berlin-R1	10.8.34.1	Cisco	CGS-MGS-AGS
Bur-isp-gw1	104.207.208.1...	Cisco	ASR 1002-X Router
CA-TOR-R1	10.8.8.8	Cisco	CGS-MGS-AGS
EDGE-234	10.99.99.171		
ELLYRENFIELDA	40.15.254.18		
ELLYRFRESNOB	40.15.254.17		

2.8.5 Duplicate Device Section

When creating or editing a NI, the visual parser and diagnosis defined under the current device can be duplicated for another device using the ***Duplicate Section*** function.

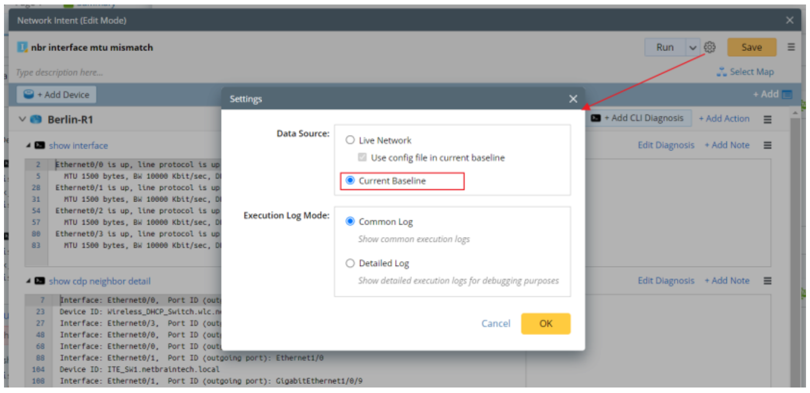


```
FastEthernet0/0 is up, line protocol is up
  Internet Address 172.24.32.225/28, Area 100
  Process ID 110, Router ID 172.24.32.225, Network Type BROADCAST, Cost: 1
Loopback80000 is up, line protocol is up
  Internet Address 172.24.255.8/32, Area 0
  Process ID 10, Router ID 172.24.255.8, Network Type LOOPBACK, Cost: 1
Serial0/1/0 is up, line protocol is up
  Internet Address 172.24.32.210/28, Area 0
  Process ID 10, Router ID 172.24.255.8, Network Type POINT_TO_POINT, Cost: 64
FastEthernet0/1 is up, line protocol is up
  Internet Address 172.24.31.195/26, Area 0
  Process ID 10, Router ID 172.24.255.8, Network Type BROADCAST, Cost: 1
```

```
Neighbor ID     Pri   State           Dead Time   Address         Interface
172.24.36.1       1   FULL/DR         00:00:38    172.24.32.226   FastEthernet0/0
172.24.255.5      0   FULL/  -        00:00:35    172.24.32.209   Serial0/1/0
172.24.31.125     1   FULL/DR         00:00:32    172.24.31.193   FastEthernet0/1
```

2.8.6 Test Run with Baseline Data

In R10.0, NI can be run with the last result data, testing the NI. R10.1 adds a new option, **Current Baseline**, for the Data source of NI besides the existing **Live Network** option.



Data Source: ( ) Live Network, ☐ Use config file in current baseline, [✓ Current Baseline]

Execution Log Mode: [✓ Common Log] Show common execution logs; ( ) Detailed Log Show detailed execution logs for debugging purposes

```
Ethernet0/0 is up, line protocol is up
  MTU 1500 bytes, BW 10000 Kbit/sec, D
Ethernet0/1 is up, line protocol is up
  MTU 1500 bytes, BW 10000 Kbit/sec, D
Ethernet0/2 is up, line protocol is up
  MTU 1500 bytes, BW 10000 Kbit/sec, D
Ethernet0/3 is up, line protocol is up
  MTU 1500 bytes, BW 10000 Kbit/sec, D
```

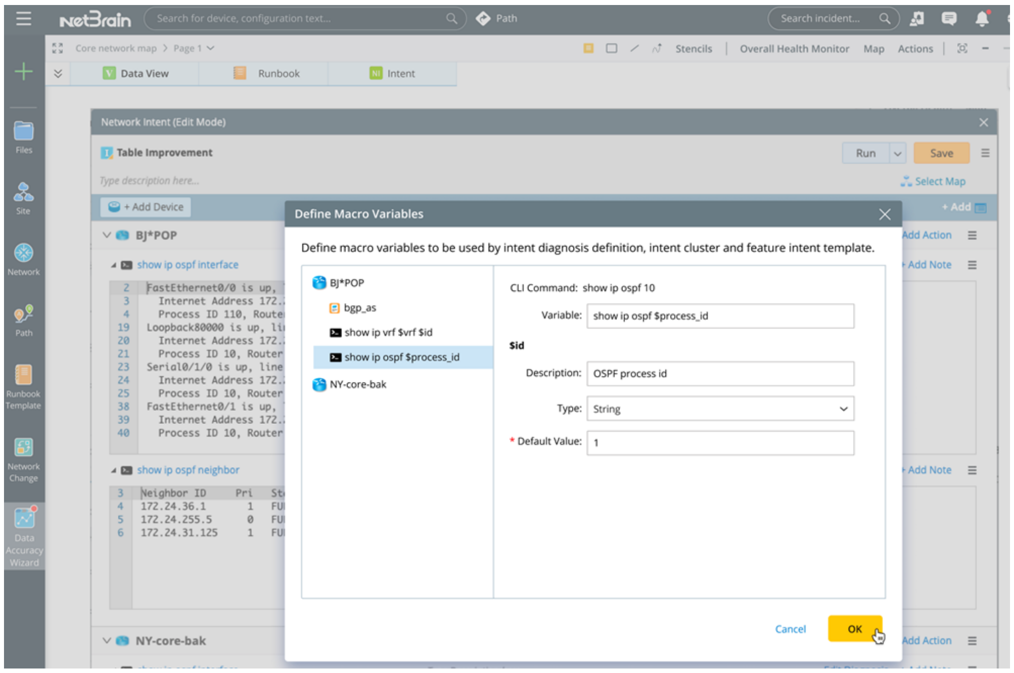
```
Interface: Ethernet0/0,  Port ID (out
Device ID: Wireless_DHCP_Switch.wlc.n
Interface: Ethernet0/3,  Port ID (out
Interface: Ethernet0/0,  Port ID (out
Interface: Ethernet0/0,  Port ID (out
Interface: Ethernet0/1,  Port ID (outgoing port): Ethernet1/0
Device ID: ITE_SW1.netbraintech.local
Interface: Ethernet0/1,  Port ID (outgoing port): GigabitEthernet1/0/9
```

2.8.7 Use Macro Variable in NI

Macro Variables can be used in NI. The Macro Variable is defined and assigned with the default value in NI edit mode. After that, Macro variables can be used in Diagnosis Conditions, Diagnosis Notes, Status Code definition, and CSV report. The default value of the Macro Variable will be used when executing the NI and exporting the report.

Users can define two types of Macro variables, the device and CLI variables:

- **CLI Variables** are Macro Variables used in CLI commands. The screenshot below gives an example using CLI Variables for the command, ***show ip ospf $process_id***.



```
FastEthernet0/0 is up,
  Internet Address 172.
  Process ID 110, Route
Loopback80000 is up, li
  Internet Address 172.
  Process ID 10, Router
Serial0/1/0 is up, line
  Internet Address 172.
  Process ID 10, Router
FastEthernet0/1 is up,
  Internet Address 172.
  Process ID 10, Router
```

```
Neighbor ID     Pri   St
172.24.36.1       1   FU
172.24.255.5      0   FU
172.24.31.125     1   FU
```

Define Macro Variables

Define macro variables to be used by intent diagnosis definition, intent cluster and feature intent template.

- BJ*POP
  - bgp_as
  - show ip vrf $vrf $id
  - show ip ospf $process_id
- NY-core-bak

CLI Command: show ip ospf 10

Variable: show ip ospf $process_id

$id

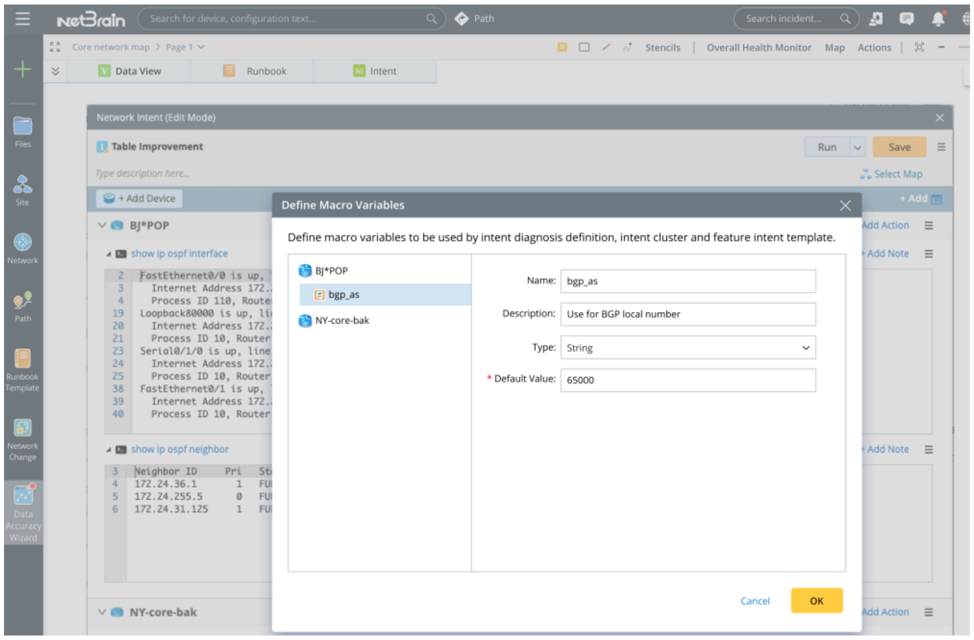
Description: OSPF process id

Type: String

* Default Value: 1

Cancel OK

- **Device Variables** are Macro Variables used on devices and can be used in NI Diagnosis Conditions, Diagnosis Notes, or CSV reports. The screenshot below gives an example using Device Variables for BGP AS number.



```
FastEthernet0/0 is up,
  Internet Address 172.
  Process ID 110, Route
Loopback80000 is up, li
  Internet Address 172.
  Process ID 10, Router
Serial0/1/0 is up, line
  Internet Address 172.
  Process ID 10, Router
FastEthernet0/1 is up,
  Internet Address 172.
  Process ID 10, Router
```

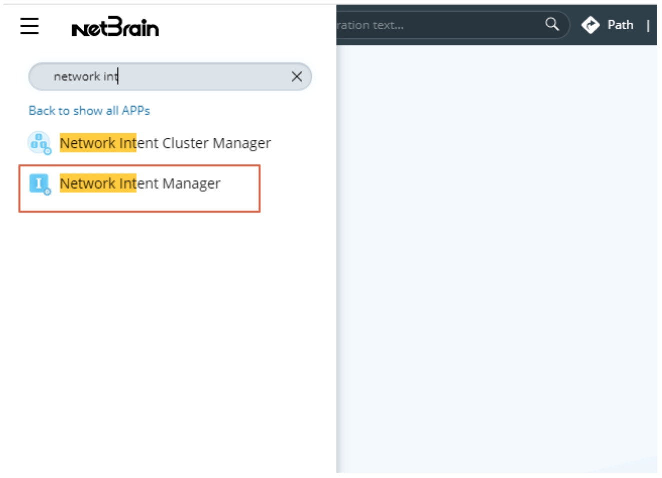
```
Neighbor ID     Pri   St
172.24.36.1       1   FU
172.24.255.5      0   FU
172.24.31.125     1   FU
```

Define macro variables to be used by intent diagnosis definition, intent cluster and feature intent template.

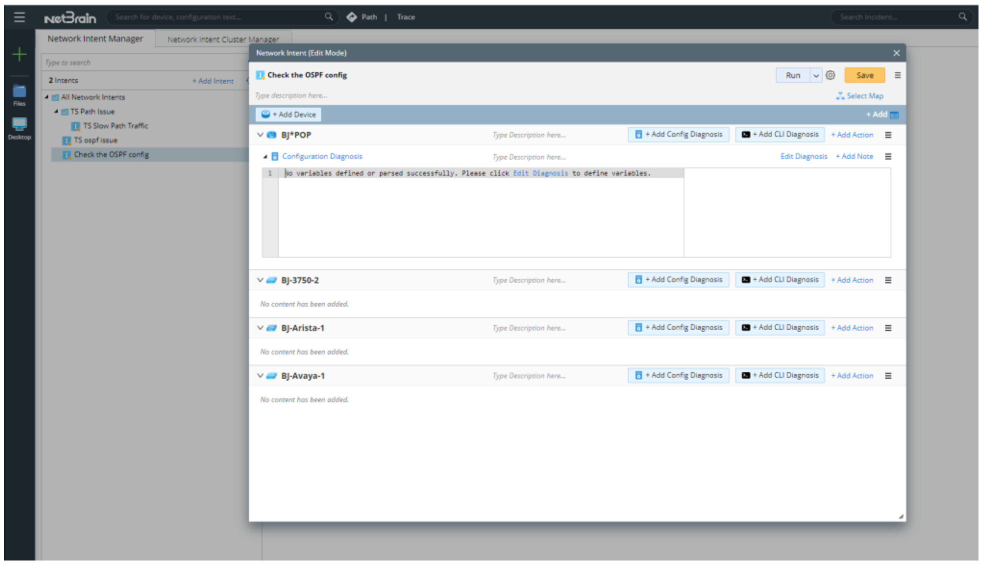
Name:	bgp_as
Description:	Use for BGP local number
Type:	String
* Default Value:	65000

2.8.8 Network Intent Manager

In the previous version, the user needs to create/edit/manage NI from the Network Intent Pane on a map, which is not convenient since only NIs applicable to the current map are displayed in this pane by default. IE10.1 adds a standalone NI manager so that users can manage NIs without opening a map.



NI Manager and NIC Manager are two tags on the same page to switch them easily. In NI Manager, all NI are managed with hierarchical folders.



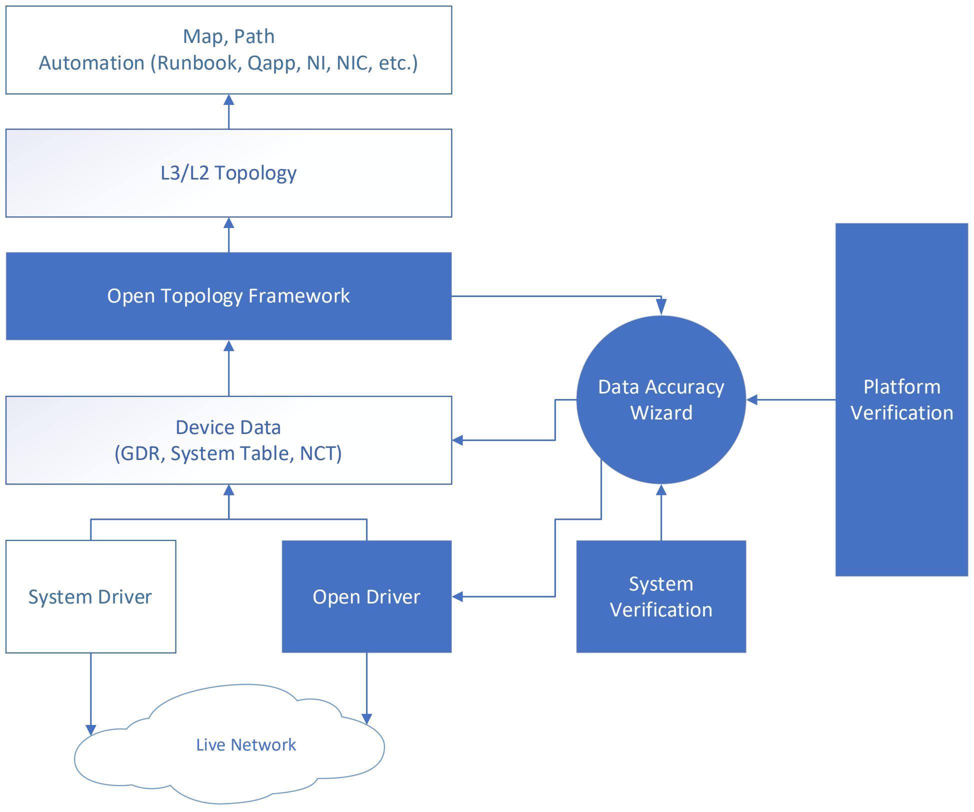
```
No variables defined or parsed successfully. Please click Edit Diagnosis to define variables.
```

3 Open Topology and Data Accuracy

NetBrain has built a complex and extendable framework to retrieve the data from the live network and create a digital twin for the network. The system driver is used to retrieve the data, a complex algorithm is developed to calculate the topology, and an intelligent path framework is created to discover the traffic path. However, due to the complexity of the network and emerging technologies, data accuracy is still common. Although we created many functions and processes to address the data accuracy issues, for example, *Tune Live Access* to take care of the device credential change, *Platform Certification* to expose all data accuracy issues once, we still have the following challenge on the data accuracy:

- No scalable solution to detect all data accuracy once. Many data accuracy issues are often unknown until users use the system to do certain tasks, such as troubleshooting a real network problem, which is certainly not a good time to resolve the data accuracy. *Platform Certification* project is a good start, but it can only be performed by the NetBrain team.
- The data accuracy issues can be caused by many functions and misconfigurations. The system can detect these issues and provide the functions to fix these issues. However, these issues are scattered in many places, such as duplicated IP manager, benchmark report, domain health report, tune live access, etc. The system does not have a central place to list all data accuracy issues and the methods to fix these issues.
- The NetBrain team resolves most data accuracy issues by modifying the system driver or adding a new driver and applying the patch of the topology algorithm. It is difficult for the user to get involved.

V10.1 provides a scalable solution to identify and fix the data accuracy issues, as illustrated by the following diagram with these new features:



- Open Topology (OT):

  The algorithm at v10.0 and earlier versions is a "closed" system, which is hard to maintain and can take a long time to fix the inaccurate data. V10.1 introduces Open Topology (OT), a new framework, to calculate the L3 and L2 topology. Besides redesigning the algorithm to improve the topology accuracy, OT is an open process to incorporate the user's input in every stage of the process and is easier to maintain. OT takes the data from the network and the user input to calculate the VLAN Group and then builds L3 and L2 topology inside each VLAN Group.

- Platform Verification (PV)

  Platform Validation (PV) checks the data accuracy by executing a set of rules called PV Rule (PVR). Besides the validation logic, a PVR will have a unique error code, error message, and recommended actions. A scheduled PV task will expose the data accuracy once, and the results are displayed in the Data Accuracy

Wizard (DAW) per device. After that, the user can view the results and take corresponding actions to fix the issue.

- System Verification (SV)

  System Verification (SV) collects all data accuracy issues caused by the system misconfiguration/failure and live access issues and displays the results in DAW with the error code, message, and recommendations.

- Data Accuracy Wizard (DAW)

  Data Accuracy Wizard (DAW) is central to addressing the data accuracy issues per device. Users can view all info related to the data accuracy, including all data errors created by PV and SV with the recommended actions, the topology and VLAN groups created by OT, Platform Verification Rules, and Open Drivers applicable to this device.

- Open Driver (OD)

  In earlier versions, the device data are retrieved and parsed by the device driver (System Driver), maintained, and updated by NetBrain. When device data is inaccurate, the user can only contact NetBrain and wait for a patch of the driver to fix the issue. IE v10.1 enables users to correct the data accuracy issues via a new feature, Open Driver (OD), which allows the user to map a variable defined by Visual Parser to a device data.

With these functions, the user flow to address the data accuracy will be:

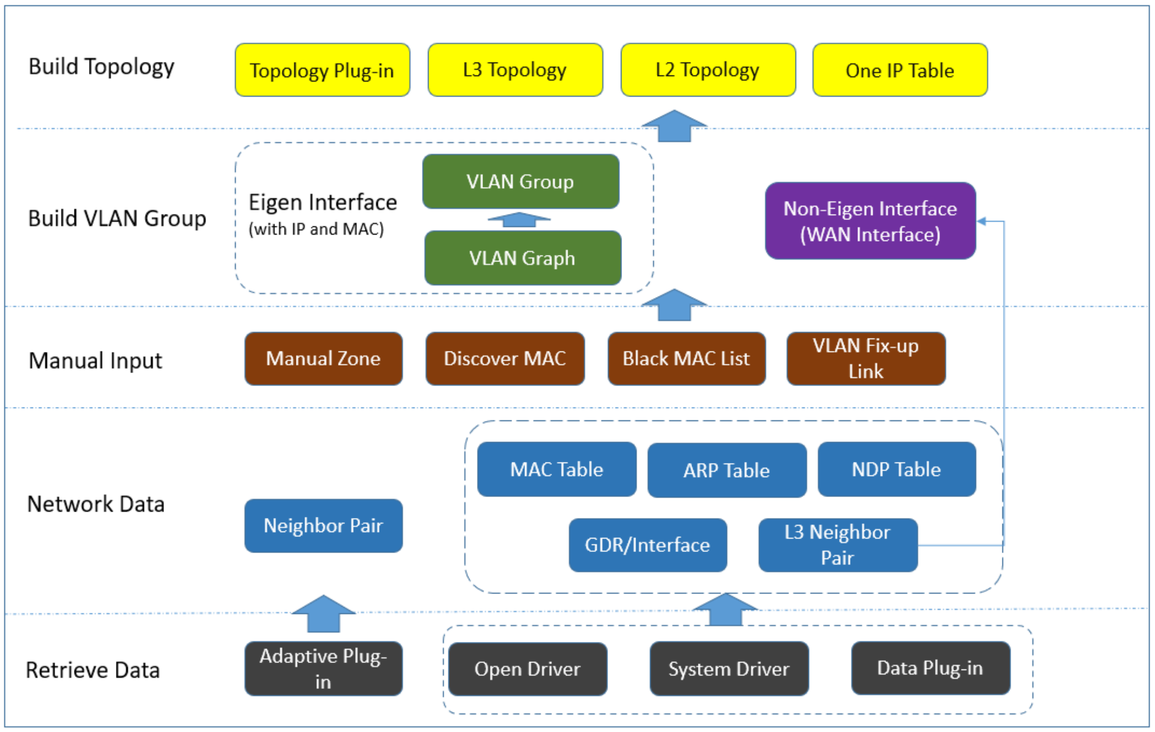
1. Schedule Run PV.
2. View all device errors in DAW, created by PV and SV.
3. Follow the recommendations to fix an error. One common method is to create an OD.

3.1 Open Topology

The topology is the foundation of the IE system. In IE 10.0 and earlier versions, we have kept the same framework to calculate L3 and L2 topology and made gradual improvements supporting emerging technologies such as virtualization and VxLAN. However, the old framework is a “closed” system, which is hard to maintain and can take a long time to fix the inaccurate data.

V10.1 introduces a new framework, Open Topology (OT), to calculate the L3 and L2 topology. Besides redesigning the algorithm to improve the topology accuracy, OT is an open process to incorporate the user's input in the different stages and is easier to maintain. OD takes the data from the network （Interface, MAC Table, ARP Table, CDP/LLDP, routing neighbors, etc.） and the user input to calculate the VLAN Group and then build L3 and L2 topology inside each VLAN Group.

The framework is illustrated in the following diagram. At each stage, the user can adjust the data or provide the manual input, for example, create an Open Driver (OD) to correct any network data, add an Adaptive Plugin to create neighbor pairs, modify the Discover and Black MAC list to adjust VLAN groups, and manually modify the topology results as a last resort.



3.1.1 Open Topology Framework

The process flow of OT is:

1. Retrieve and parse the data from the network with the System Driver. Then, use the Open Driver and Data Plug-in to fix any inaccurate data.
2. Execute the Adaptive Plugins for the special technologies such as VxLAN and OTV to generate Neighbor Pair Table.
3. Calculate the VLAN Group. Use the network data combined with the user input Discover MAC, Black MAC, and VLAN Fix-up Link as the input.
4. Generate Final Zone with VLAN Group and the user input Manual Zone.
5. Calculate L3 Topo for each Final Zone, calculate L2 Topo and One IP Table for each VLAN Group.
6. Add the Neighbor Pair to the topology results.
7. Run MPLS Plug-in and other Topology Plug-in to correct or supplement the topology results.

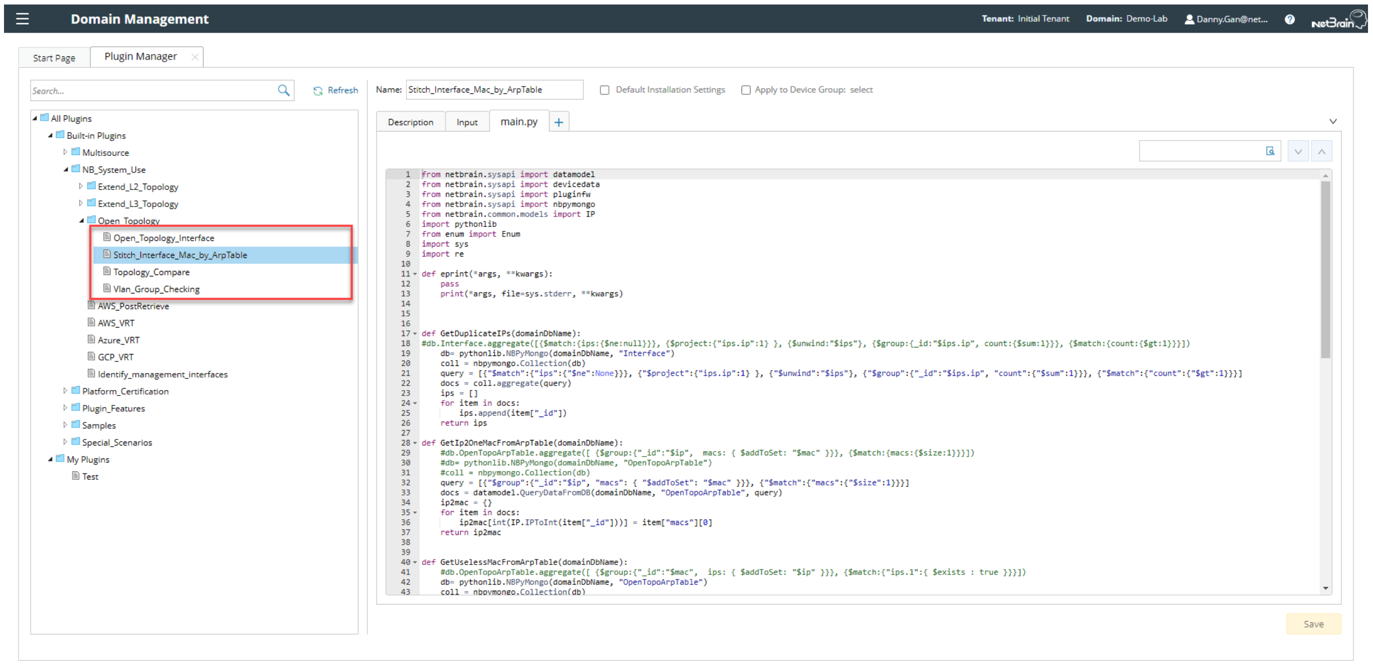
3.1.1.1 VLAN Group

VLAN Group is an L2 Broadcast Domain, including L3 IP Interface, L2 Switchport, and End Point (End System and MAC Device). It is often associated with a business unit, for example, Dev VLAN, QA VLAN, etc. The IP addresses in a VLAN Group should not be duplicated except HA (High Availability) and FHRP (First Hop Redundancy Protocol).

3.1.1.2 Data and Adaptive Plug-in

Each device in the system is assigned to a System Driver (SD), which defines how to retrieve and parse the built-in data for a device type. Since SD is maintained by NetBrain and published to all customers, it may take a while to modify SD to fix any data accuracy issues. Instead, the user can create an Open Driver or Data Plug-in to fix the issue quickly. For example, the interface of an ARP table is empty.

Adaptive Plugin is used to create the neighbor pairs for special technologies such as VxLAN and OTV, keeping the core algorithm simple and efficient and supporting certain technologies without a new release.



Name: Stitch_Interface_Mac_by_ArpTable

Default Installation Settings

Apply to Device Group: select

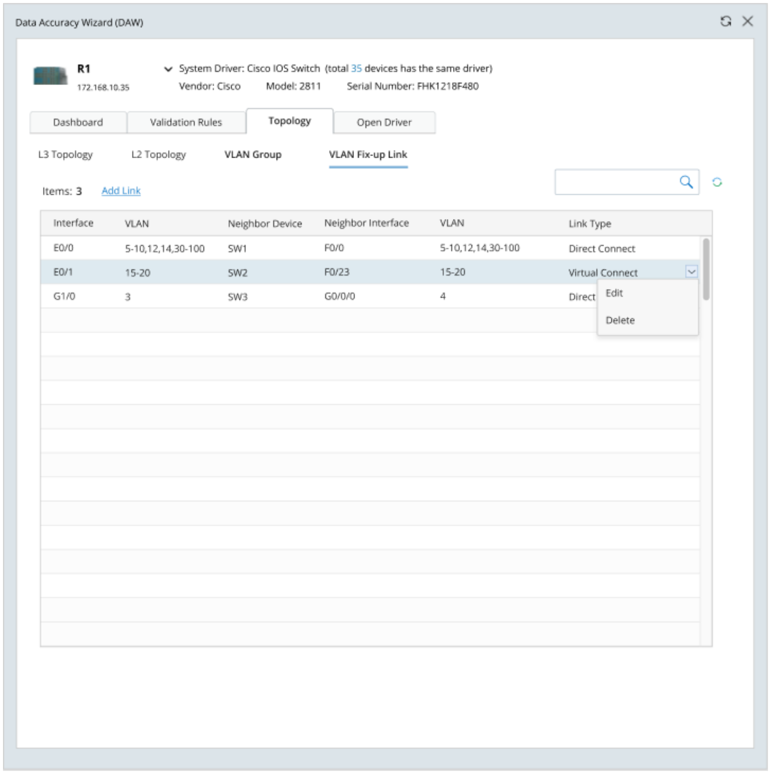
```python
from netbrain.sysapi import datamodel
from netbrain.sysapi import devicedata
from netbrain.sysapi import pluginfw
from netbrain.sysapi import nbpymongo
from netbrain.common.models import IP
import pythonlib
from enum import Enum
import sys
import re

def eprint(*args, **kwargs):
    pass
    print(*args, file=sys.stderr, **kwargs)



def GetDuplicateIPs(domainDbName):
#db.Interface.aggregate([{$match:{ips:{$ne:null}}}, {$project:{"ips.ip":1} }, {$unwind:"$ips"}, {$group:{_id:"$ips.ip", count:{$sum:1}}}, {$match:{count:{$gt:1}}}])
    db= pythonlib.NBPyMongo(domainDbName, "Interface")
    coll = nbpymongo.Collection(db)
    query = [{"$match":{"ips":{"$ne":None}}}, {"$project":{"ips.ip":1} }, {"$unwind":"$ips"}, {"$group":{"_id":"$ips.ip", "count":{"$sum":1}}}, {"$match":{"count":{"$gt":1}}}]
    docs = coll.aggregate(query)
    ips = []
    for item in docs:
        ips.append(item["_id"])
    return ips

def GetIp2OneMacFromArpTable(domainDbName):
    #db.OpenTopoArpTable.aggregate([ {$group:{"_id":"$ip",  macs: { $addToSet: "$mac" }}}, {$match:{macs:{$size:1}}}])
    #db= pythonlib.NBPyMongo(domainDbName, "OpenTopoArpTable")
    #coll = nbpymongo.Collection(db)
    query = [{"$group":{"_id":"$ip", "macs": { "$addToSet": "$mac" }}}, {"$match":{"macs":{"$size":1}}}]
    docs = datamodel.QueryDataFromDB(domainDbName, "OpenTopoArpTable", query)
    ip2mac = {}
    for item in docs:
        ip2mac[int(IP.IPToInt(item["_id"]))] = item["macs"][0]
    return ip2mac


def GetUselessMacFromArpTable(domainDbName):
    #db.OpenTopoArpTable.aggregate([ {$group:{"_id":"$mac",  ips: { $addToSet: "$ip" }}}, {$match:{"ips.1":{ $exists : true }}}])
    db= pythonlib.NBPyMongo(domainDbName, "OpenTopoArpTable")
    coll = nbpymongo.Collection(db)
```

3.1.1.3 Discover and Black MAC

The user can define the MAC addresses that must be used to create a VLAN group as the Discover MAC and those that must NOT be used as the Black MAC. Discover MAC addresses are those stable MAC addresses used often in communication. Discover MAC address must be unique across the whole network. In contrast, Black MAC addresses are not used in real traffic and may cause the miscalculation of VLAN groups. Both addresses can be edited in Domain Management.

3.1.1.4 VLAN Fix-up Link

The user can manually add the links between device interfaces and VLAN info on the Data Accuracy Wizard (DAW) pane.



Data Accuracy Wizard (DAW)

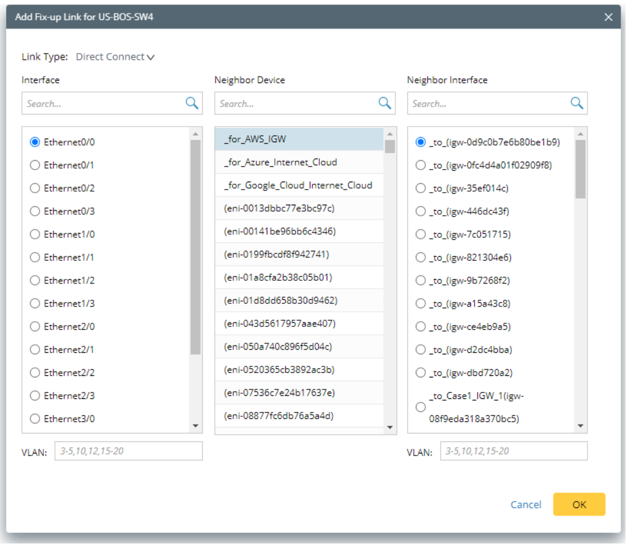
R1
172.168.10.35

System Driver: Cisco IOS Switch (total 35 devices has the same driver)
Vendor: Cisco Model: 2811 Serial Number: FHK1218F480

Items: 3 Add Link

Interface	VLAN	Neighbor Device	Neighbor Interface	VLAN	Link Type
E0/0	5-10,12,14,30-100	SW1	F0/0	5-10,12,14,30-100	Direct Connect
E0/1	15-20	SW2	F0/23	15-20	Virtual Connect
G1/0	3	SW3	G0/0/0	4	Direct

There are two types of links, Direct Connect, which will affect both VLAN group and L2 topology, and Virtual Connect, which only affects the calculation of VLAN groups. The Virtual Connect is designed to support Overlay technology such as OTV and VxLAN.



Add Fix-up Link for US-BOS-SW4

Link Type: Direct Connect

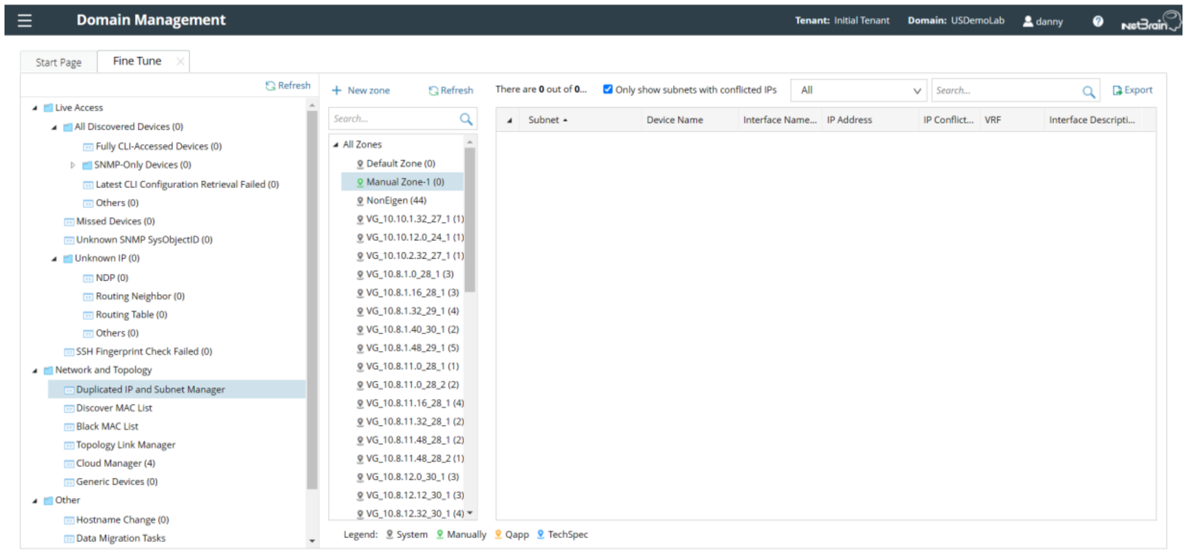
Interface	Neighbor Device	Neighbor Interface
Ethernet0/0	_for_AWS_IGW	_to_(igw-0d9c0b7e6b80be1b9)
Ethernet0/1	_for_Azure_Internet_Cloud	_to_(igw-0fc4d4a01f02909f8)
Ethernet0/2	_for_Google_Cloud_Internet_Cloud	_to_(igw-35ef014c)
Ethernet0/3	(eni-0013dbbc77e3bc97c)	_to_(igw-446dc43f)
Ethernet1/0	(eni-00141be96bb6c4346)	_to_(igw-7c051715)
Ethernet1/1	(eni-0199fbcdf8f942741)	_to_(igw-821304e6)
Ethernet1/2	(eni-01a8cfa2b38c05b01)	_to_(igw-9b7268f2)
Ethernet1/3	(eni-01d8dd658b30d9462)	_to_(igw-a15a43c8)
Ethernet2/0	(eni-043d5617957aae407)	_to_(igw-ce4eb9a5)
Ethernet2/1	(eni-050a740c896f5d04c)	_to_(igw-d2dc4bba)
Ethernet2/2	(eni-0520365cb3892ac3b)	_to_(igw-dbd720a2)
Ethernet2/3	(eni-07536c7e24b17637e)	_to_Case1_IGW_1(igw-08f9eda318a370bc5)
Ethernet3/0	(eni-08877fc6db76a5a4d)	

VLAN: 3-5,10,12,15-20

VLAN: 3-5,10,12,15-20

3.1.1.5 Manual Zone

V10.1 keeps Manual Zone to solve the duplicated IP address issues and adjust the L3 topology results.

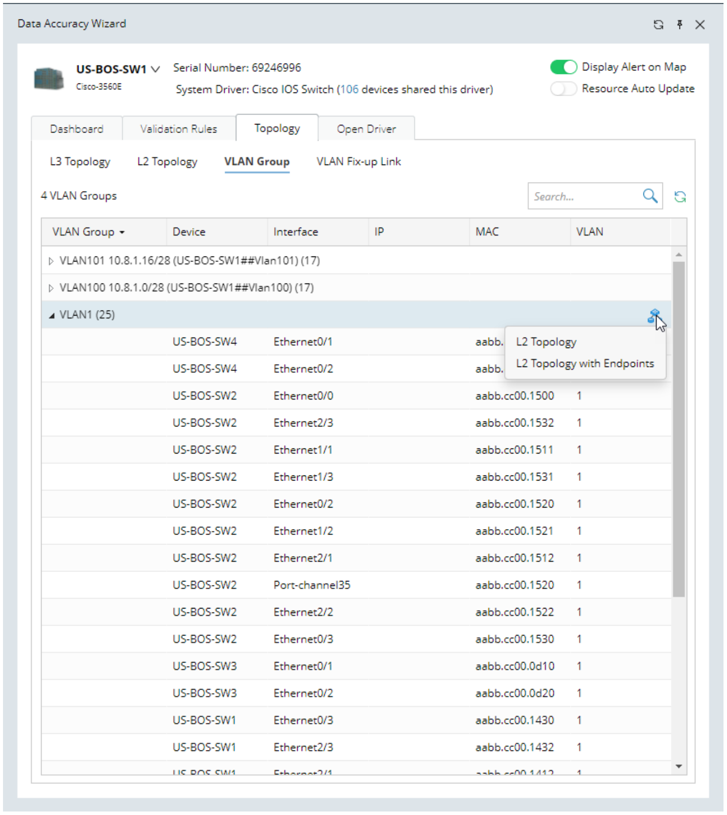


3.1.2 Open Topo Output

The OT results can be viewed per device in DAW.

3.1.2.1 View and Map VLAN Group

Under the Topology tag of DAW, a user can view all VLAN groups this device belongs to, view VLAN group details, and create the L2 topology map for a VLAN group. In addition, the system provides an option to include the end systems in the VLAN map.


Data Accuracy Wizard

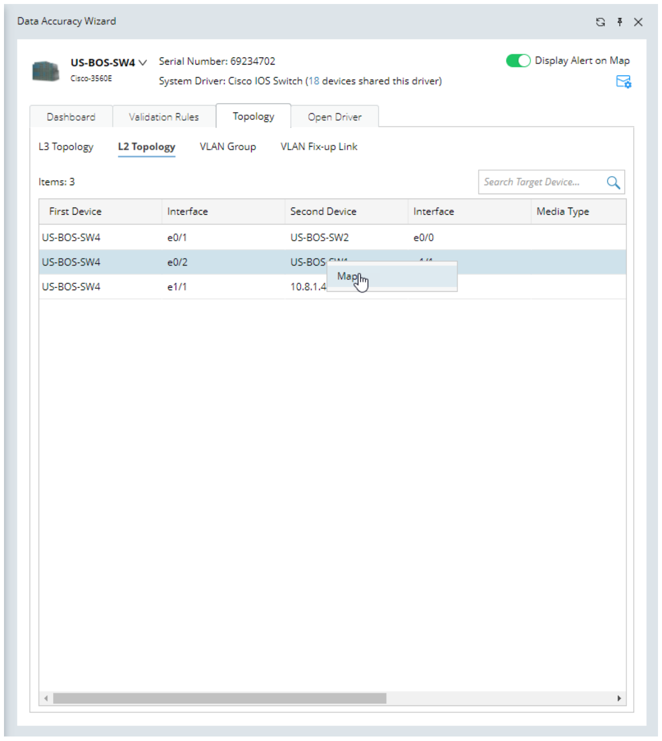
US-BOS-SW1 Cisco-3560E
Serial Number: 69246996
System Driver: Cisco IOS Switch (106 devices shared this driver)

4 VLAN Groups

VLAN Group	Device	Interface	IP	MAC	VLAN
VLAN101 10.8.1.16/28 (US-BOS-SW1##Vlan101) (17)					
VLAN100 10.8.1.0/28 (US-BOS-SW1##Vlan100) (17)					
VLAN1 (25)					
	US-BOS-SW4	Ethernet0/1		aabb.	
	US-BOS-SW4	Ethernet0/2		aabb.	
	US-BOS-SW2	Ethernet0/0		aabb.cc00.1500	1
	US-BOS-SW2	Ethernet2/3		aabb.cc00.1532	1
	US-BOS-SW2	Ethernet1/1		aabb.cc00.1511	1
	US-BOS-SW2	Ethernet1/3		aabb.cc00.1531	1
	US-BOS-SW2	Ethernet0/2		aabb.cc00.1520	1
	US-BOS-SW2	Ethernet1/2		aabb.cc00.1521	1
	US-BOS-SW2	Ethernet2/1		aabb.cc00.1512	1
	US-BOS-SW2	Port-channel35		aabb.cc00.1520	1
	US-BOS-SW2	Ethernet2/2		aabb.cc00.1522	1
	US-BOS-SW2	Ethernet0/3		aabb.cc00.1530	1
	US-BOS-SW3	Ethernet0/1		aabb.cc00.0d10	1
	US-BOS-SW3	Ethernet0/2		aabb.cc00.0d20	1
	US-BOS-SW1	Ethernet0/3		aabb.cc00.1430	1
	US-BOS-SW1	Ethernet2/3		aabb.cc00.1432	1

3.1.2.2 View L3 and L2 Topology

At DAW, a user can view all L3 and L2 links of a device, map one or multiple topology links. Here we display the point-to-point (P2P) full mesh connections instead of the connection to the media. In addition, more details are displayed for a topology link, such as the source and updated time, to debug the data accuracy issue.


Data Accuracy Wizard

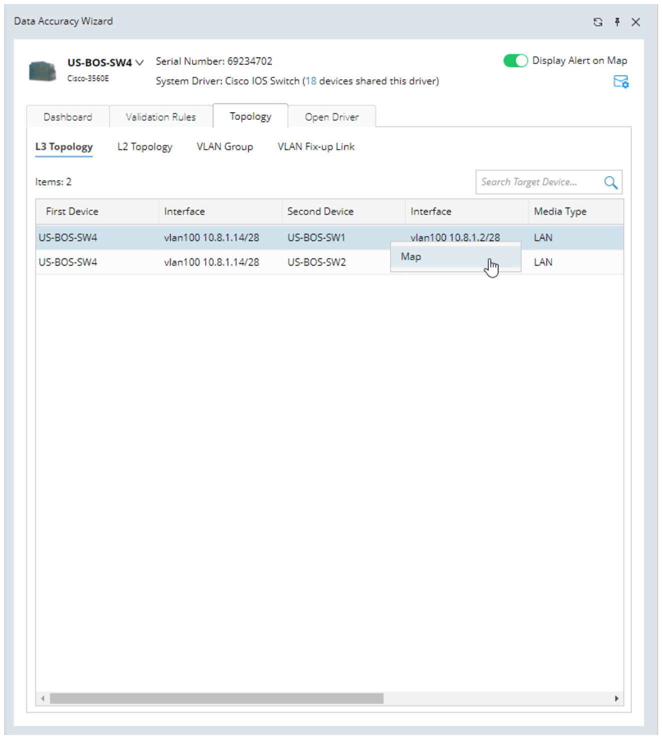
US-BOS-SW4 Cisco-3560E — Serial Number: 69234702 — System Driver: Cisco IOS Switch (18 devices shared this driver) — Display Alert on Map

Dashboard | Validation Rules | Topology | Open Driver

L3 Topology | **L2 Topology** | VLAN Group | VLAN Fix-up Link

Items: 3

First Device	Interface	Second Device	Interface	Media Type
US-BOS-SW4	e0/1	US-BOS-SW2	e0/0	
US-BOS-SW4	e0/2	US-BOS		
US-BOS-SW4	e1/1	10.8.1.4		


Data Accuracy Wizard

US-BOS-SW4 Cisco-3560E — Serial Number: 69234702 — System Driver: Cisco IOS Switch (18 devices shared this driver) — Display Alert on Map

Dashboard | Validation Rules | Topology | Open Driver

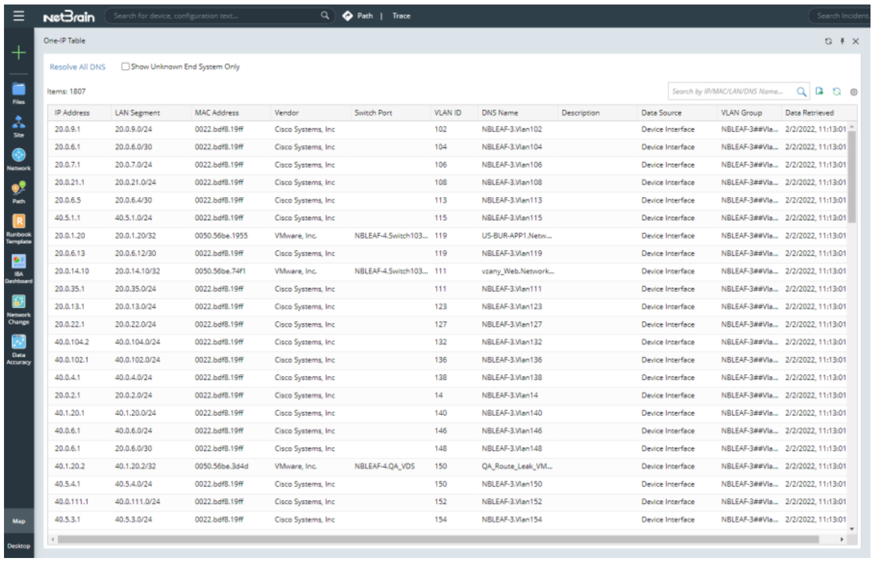
L3 Topology | L2 Topology | VLAN Group | VLAN Fix-up Link

Items: 2

First Device	Interface	Second Device	Interface	Media Type
US-BOS-SW4	vlan100 10.8.1.14/28	US-BOS-SW1	vlan100 10.8.1.2/28	LAN
US-BOS-SW4	vlan100 10.8.1.14/28	US-BOS-SW2		LAN

3.1.2.3 One IP Table

V10.1 adds two fields, VLAN Group and VLAN ID, in the One IP table. VLAN Group Name or ID is displayed under VLAN Group, and the VLAN ID is VLAN Number.



One-IP Table

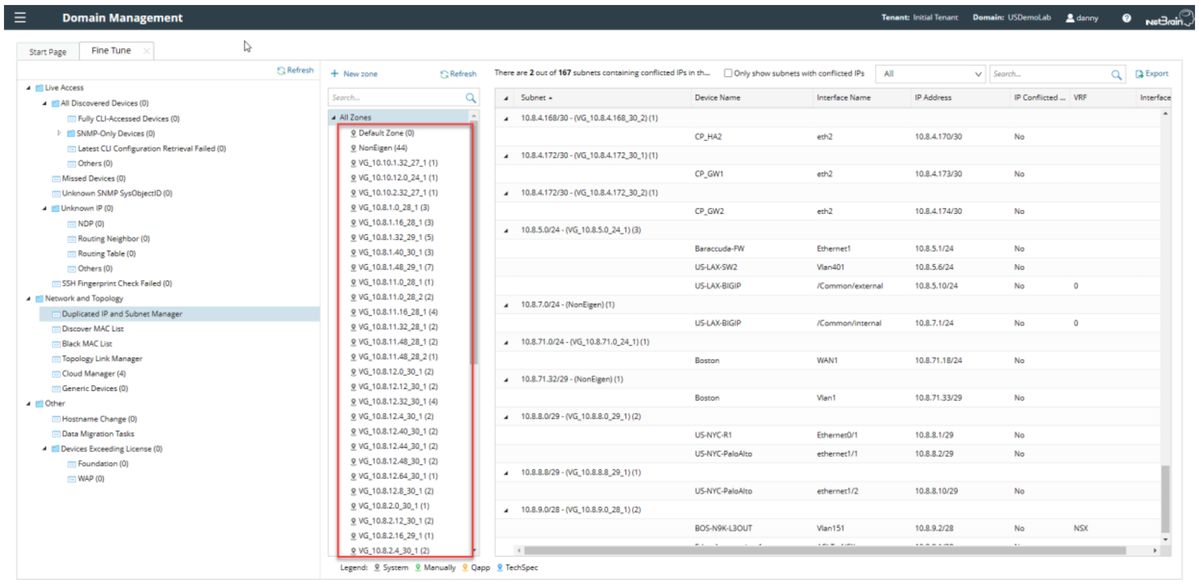
Resolve All DNS   ☐ Show Unknown End System Only

Items: 1807

IP Address	LAN Segment	MAC Address	Vendor	Switch Port	VLAN ID	DNS Name	Description	Data Source	VLAN Group	Data Retrieved
20.0.9.1	20.0.9.0/24	0022.bdf8.19ff	Cisco Systems, Inc		102	NBLEAF-3.Vlan102		Device Interface	NBLEAF-3##Vla...	2/2/2022, 11:13:01
20.0.6.1	20.0.6.0/30	0022.bdf8.19ff	Cisco Systems, Inc		104	NBLEAF-3.Vlan104		Device Interface	NBLEAF-3##Vla...	2/2/2022, 11:13:01
20.0.7.1	20.0.7.0/24	0022.bdf8.19ff	Cisco Systems, Inc		106	NBLEAF-3.Vlan106		Device Interface	NBLEAF-3##Vla...	2/2/2022, 11:13:01
20.0.21.1	20.0.21.0/24	0022.bdf8.19ff	Cisco Systems, Inc		108	NBLEAF-3.Vlan108		Device Interface	NBLEAF-3##Vla...	2/2/2022, 11:13:01
20.0.6.5	20.0.6.4/30	0022.bdf8.19ff	Cisco Systems, Inc		113	NBLEAF-3.Vlan113		Device Interface	NBLEAF-3##Vla...	2/2/2022, 11:13:01
40.5.1.1	40.5.1.0/24	0022.bdf8.19ff	Cisco Systems, Inc		115	NBLEAF-3.Vlan115		Device Interface	NBLEAF-3##Vla...	2/2/2022, 11:13:01
20.0.1.20	20.0.1.20/32	0050.56be.1955	VMware, Inc.	NBLEAF-4.Switch103...	119	US-BUR-APP1.Netw...		Device Interface	NBLEAF-3##Vla...	2/2/2022, 11:13:01
20.0.6.13	20.0.6.12/30	0022.bdf8.19ff	Cisco Systems, Inc		119	NBLEAF-3.Vlan119		Device Interface	NBLEAF-3##Vla...	2/2/2022, 11:13:01
20.0.14.10	20.0.14.10/32	0050.56be.74f1	VMware, Inc.	NBLEAF-4.Switch103...	111	vzany_Web.Network...		Device Interface	NBLEAF-3##Vla...	2/2/2022, 11:13:01
20.0.35.1	20.0.35.0/24	0022.bdf8.19ff	Cisco Systems, Inc		111	NBLEAF-3.Vlan111		Device Interface	NBLEAF-3##Vla...	2/2/2022, 11:13:01
20.0.13.1	20.0.13.0/24	0022.bdf8.19ff	Cisco Systems, Inc		123	NBLEAF-3.Vlan123		Device Interface	NBLEAF-3##Vla...	2/2/2022, 11:13:01
20.0.22.1	20.0.22.0/24	0022.bdf8.19ff	Cisco Systems, Inc		127	NBLEAF-3.Vlan127		Device Interface	NBLEAF-3##Vla...	2/2/2022, 11:13:01
40.0.104.2	40.0.104.0/24	0022.bdf8.19ff	Cisco Systems, Inc		132	NBLEAF-3.Vlan132		Device Interface	NBLEAF-3##Vla...	2/2/2022, 11:13:01
40.0.102.1	40.0.102.0/24	0022.bdf8.19ff	Cisco Systems, Inc		136	NBLEAF-3.Vlan136		Device Interface	NBLEAF-3##Vla...	2/2/2022, 11:13:01
40.0.4.1	40.0.4.0/24	0022.bdf8.19ff	Cisco Systems, Inc		138	NBLEAF-3.Vlan138		Device Interface	NBLEAF-3##Vla...	2/2/2022, 11:13:01
20.0.2.1	20.0.2.0/24	0022.bdf8.19ff	Cisco Systems, Inc		14	NBLEAF-3.Vlan14		Device Interface	NBLEAF-3##Vla...	2/2/2022, 11:13:01
40.1.20.1	40.1.20.0/24	0022.bdf8.19ff	Cisco Systems, Inc		140	NBLEAF-3.Vlan140		Device Interface	NBLEAF-3##Vla...	2/2/2022, 11:13:01
40.0.6.1	40.0.6.0/24	0022.bdf8.19ff	Cisco Systems, Inc		146	NBLEAF-3.Vlan146		Device Interface	NBLEAF-3##Vla...	2/2/2022, 11:13:01
20.0.6.1	20.0.6.0/30	0022.bdf8.19ff	Cisco Systems, Inc		148	NBLEAF-3.Vlan148		Device Interface	NBLEAF-3##Vla...	2/2/2022, 11:13:01
40.1.20.2	40.1.20.2/32	0050.56be.3d4d	VMware, Inc.	NBLEAF-4.QA_VDS	150	QA_Route_Leak_VM...		Device Interface	NBLEAF-3##Vla...	2/2/2022, 11:13:01
40.5.4.1	40.5.4.0/24	0022.bdf8.19ff	Cisco Systems, Inc		150	NBLEAF-3.Vlan150		Device Interface	NBLEAF-3##Vla...	2/2/2022, 11:13:01
40.0.111.1	40.0.111.0/24	0022.bdf8.19ff	Cisco Systems, Inc		152	NBLEAF-3.Vlan152		Device Interface	NBLEAF-3##Vla...	2/2/2022, 11:13:01
40.5.3.1	40.5.3.0/24	0022.bdf8.19ff	Cisco Systems, Inc		154	NBLEAF-3.Vlan154		Device Interface	NBLEAF-3##Vla...	2/2/2022, 11:13:01

3.1.2.4 VLAN Group Based Zone

Most duplicated IP addresses are resolved by the auto-created VLAN Group Zone, as shown in the Duplicate IP and Subnet Manager.



All Zones
- Default Zone (0)
- NonEigen (44)
- VG_10.10.1.32_27_1 (1)
- VG_10.10.12.0_24_1 (1)
- VG_10.10.2.32_27_1 (1)
- VG_10.8.1.0_28_1 (3)
- VG_10.8.1.16_28_1 (3)
- VG_10.8.1.32_29_1 (5)
- VG_10.8.1.40_30_1 (3)
- VG_10.8.1.48_29_1 (7)
- VG_10.8.11.0_28_1 (1)
- VG_10.8.11.0_28_2 (2)
- VG_10.8.11.16_28_1 (4)
- VG_10.8.11.32_28_1 (2)
- VG_10.8.11.48_28_1 (2)
- VG_10.8.11.48_28_2 (1)
- VG_10.8.12.0_30_1 (2)
- VG_10.8.12.12_30_1 (2)
- VG_10.8.12.32_30_1 (4)
- VG_10.8.12.4_30_1 (2)
- VG_10.8.12.40_30_1 (2)
- VG_10.8.12.44_30_1 (2)
- VG_10.8.12.48_30_1 (2)
- VG_10.8.12.64_30_1 (1)
- VG_10.8.12.8_30_1 (2)
- VG_10.8.2.0_30_1 (1)
- VG_10.8.2.12_30_1 (2)
- VG_10.8.2.16_29_1 (1)
- VG_10.8.2.4_30_1 (2)

There are **2** out of **167** subnets containing conflicted IPs in th...

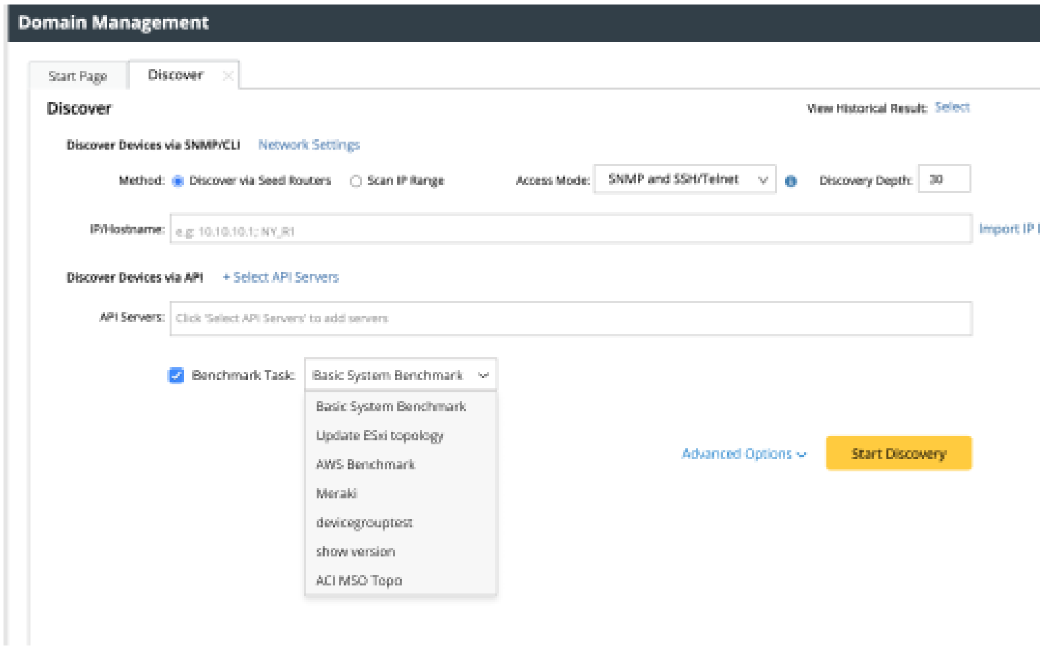
Subnet	Device Name	Interface Name	IP Address	IP Conflicted ...	VRF	Interface
10.8.4.168/30 - (VG_10.8.4.168_30_2) (1)						
	CP_HA2	eth2	10.8.4.170/30	No		
10.8.4.172/30 - (VG_10.8.4.172_30_1) (1)						
	CP_GW1	eth2	10.8.4.173/30	No		
10.8.4.172/30 - (VG_10.8.4.172_30_2) (1)						
	CP_GW2	eth2	10.8.4.174/30	No		
10.8.5.0/24 - (VG_10.8.5.0_24_1) (3)						
	Baraccuda-FW	Ethernet1	10.8.5.1/24	No		
	US-LAX-SW2	Vlan401	10.8.5.6/24	No		
	US-LAX-BIGIP	/Common/external	10.8.5.10/24	No	0	
10.8.7.0/24 - (NonEigen) (1)						
	US-LAX-BIGIP	/Common/internal	10.8.7.1/24	No	0	
10.8.71.0/24 - (VG_10.8.71.0_24_1) (1)						
	Boston	WAN1	10.8.71.18/24	No		
10.8.71.32/29 - (NonEigen) (1)						
	Boston	Vlan1	10.8.71.33/29	No		
10.8.8.0/29 - (VG_10.8.8.0_29_1) (2)						
	US-NYC-R1	Ethernet0/1	10.8.8.1/29	No		
	US-NYC-PaloAlto	ethernet1/1	10.8.8.2/29	No		
10.8.8.8/29 - (VG_10.8.8.8_29_1) (1)						
	US-NYC-PaloAlto	ethernet1/2	10.8.8.10/29	No		
10.8.9.0/28 - (VG_10.8.9.0_28_1) (2)						
	BOS-N9K-L3OUT	Vlan151	10.8.9.2/28	No	NSX	

3.1.3 Other Related Adjustments

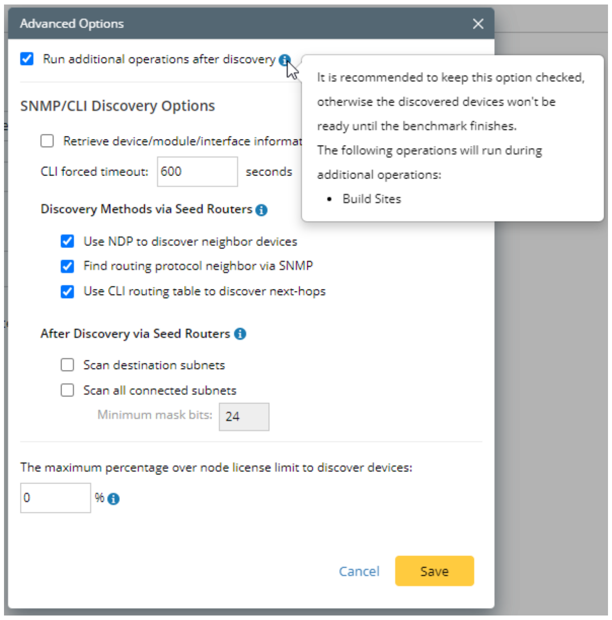
3.1.3.1 Discovery

The OT framework uses more network data than the old algorithm to build the VLAN group and then L3 and L2 topology. To keep the discovery process simple, the system does not build the topology in the discovery process. Instead, the topology will be built in the system benchmark. Therefore, discovery UI is adjusted as follows:

- Add a link to run the benchmark task in the manual Discovery UI and Domain Setup Wizard. The user can select the predefined benchmark tasks to run. This option is not available for the scheduled Discovery.



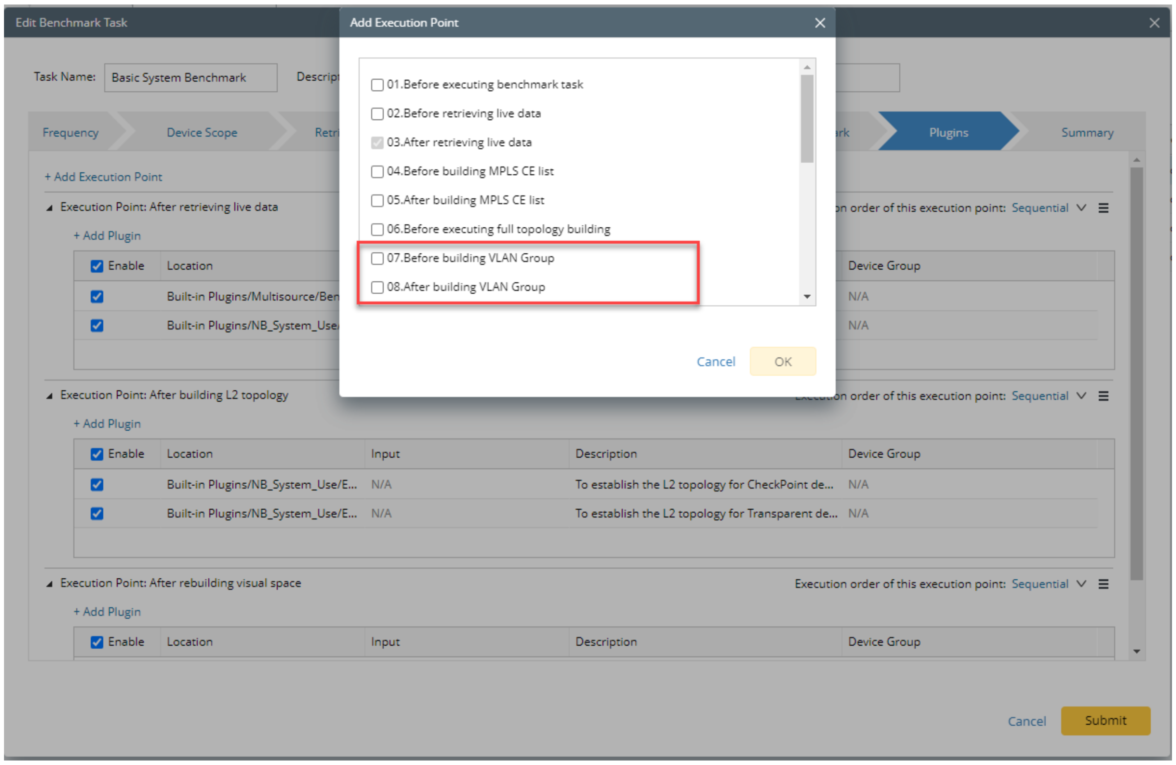
▪ Remove all operations except Build Site in Run Additional Operations After Discovery at the Advanced Options.


Advanced Options
Run additional operations after discovery
SNMP/CLI Discovery Options
Retrieve device/module/interface informat
CLI forced timeout: 600 seconds
Discovery Methods via Seed Routers
Use NDP to discover neighbor devices
Find routing protocol neighbor via SNMP
Use CLI routing table to discover next-hops
After Discovery via Seed Routers
Scan destination subnets
Scan all connected subnets
Minimum mask bits: 24
The maximum percentage over node license limit to discover devices:
0 %
Cancel
Save

It is recommended to keep this option checked, otherwise the discovered devices won't be ready until the benchmark finishes.
The following operations will run during additional operations:
- Build Sites

3.1.3.2 Benchmark Setting

In the Add Execution Point of the Benchmark task, add two entries before building VLAN Group and after building VLAN Group. The user can add the Data and Adaptive plugins to fix the topology issues.



Edit Benchmark Task

Task Name: Basic System Benchmark

Frequency | Device Scope | Retri… | …rk | Plugins | Summary

+ Add Execution Point

Execution Point: After retrieving live data — Execution order of this execution point: Sequential

+ Add Plugin

Enable	Location	Input	Description	Device Group
✓	Built-in Plugins/Multisource/Ben…			N/A
✓	Built-in Plugins/NB_System_Use…			N/A

Execution Point: After building L2 topology — Execution order of this execution point: Sequential

+ Add Plugin

Enable	Location	Input	Description	Device Group
✓	Built-in Plugins/NB_System_Use/E...	N/A	To establish the L2 topology for CheckPoint de...	N/A
✓	Built-in Plugins/NB_System_Use/E...	N/A	To establish the L2 topology for Transparent de...	N/A

Execution Point: After rebuilding visual space — Execution order of this execution point: Sequential

+ Add Plugin

Enable	Location	Input	Description	Device Group

Cancel | Submit

Add Execution Point

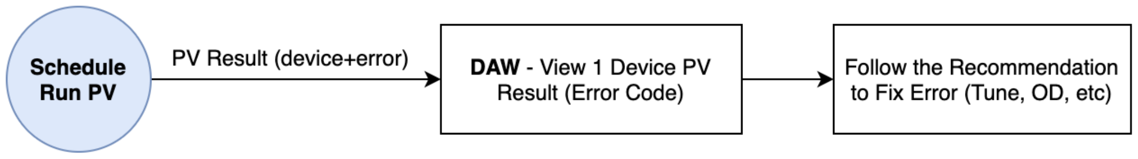
- [ ] 01.Before executing benchmark task
- [ ] 02.Before retrieving live data
- [✓] 03.After retrieving live data
- [ ] 04.Before building MPLS CE list
- [ ] 05.After building MPLS CE list
- [ ] 06.Before executing full topology building
- [ ] 07.Before building VLAN Group
- [ ] 08.After building VLAN Group

Cancel | OK

3.2 Platform Validation

Platform Validation (PV) checks the data accuracy by executing a set of rules called PV Rule (PVR). A PVR can be as simple as a device management IP address must be not empty and as complex as that a physical interface (identified by its name) with an IP address configured must not have VLAN configured. Besides the validation logic, a PVR will have a unique error code, error message, and recommended actions. PV results are displayed in the Data Accuracy Wizard (DAW) per device. NetBrain system administrator or any end-user can view the results and take corresponding actions to fix the issue.

The main workflow is: the administrator set up a scheduled task to execute PV against all devices in the domain; all NetBrain users view PV errors and warnings on DAW and follow the recommended actions to fix the issue.


Schedule Run PV
PV Result (device+error)
DAW - View 1 Device PV Result (Error Code)
Follow the Recommendation to Fix Error (Tune, OD, etc)

NetBrain Platform team has already built a rich set of PVR packaged into the V10.1. However, the network is complex, and we will run into new technology and issue. Therefore, we will continue to develop and optimize the PVR. New PVRs and the corresponding System Driver, Open Driver, or other resources will be published to the customer via Knowledge Cloud (KC).

3.2.1 Define PVR

PVR is created only by NetBrain Platform Engineers and published to the customer via KC. The PVR is defined by a Python script file (PV file). A PVR has the following components:

- **Name**: a unique name to identify the PV file name.
- **Description**: optional, describing what this PVR is about.
- **Qualification**: defining what devices this PVR can be applied to, same as the general Qualification definition. For example, you can define a PVR only applicable to a certain device type, such as the Cisco IOS router or a device with BGP configured.
- **Check Logic**: the logic to implement PVR. The logic can be simple, such as checking whether a device management IP address (the device parameter, sometimes also called GDR) is empty. If so, create an error.

The error is defined globally, which includes:

- **Error Code**: a unique hexadecimal number to identify the error.
- **Error Type**: the category of this error.
- **Short Message**: describing the error, e.g., the management IP address is empty.
- **Long Message**: describe the interface error in detail.
- **Severity**: can be Error, Warning, Information.

- **Recommendation**: each error code corresponds to multiple recommendations on how to solve the problem when such an error occurs. The common recommendation is listed in the following table.

Recommendation	Description
Create an Open Driver	To create an Open Driver to fix this issue.
Update an Open Driver	Update an existing Open Driver.
Tune Live Access	Open the device setting for the user to tune.
Update System Driver	The system will create a beta System Driver, which is enabled in the IE system.
View Original Data	The link will open the UI to view the original data, such as a system table or an NCT table.
Contact NetBrain	Send an email to NetBrain support.

The errors are listed in DAW per device. In addition, the recommendation has a link to the corresponding action UI.

3.2.2 PVR Examples

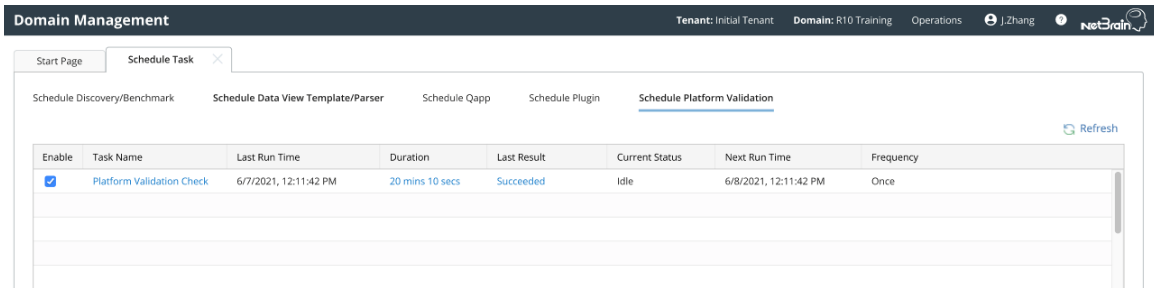
PVR can be used to validate the device/interface/module property, system table, NCT table, and Overall Health Monitoring data.

Category	Check Logic	Error Message	Recommendation
Device Property	***Management IP*** is not empty.	The Management IP is empty.	Please set the management IP in the Shared Device Settings or Contact NetBrain.
Device Property	A device supporting CLI Configurations has SNMP created configurations.	Failed to retrieve CLI configuration.	Please Contact NetBrain.

Device Property	Interfaces in the route table should exist in the domain.	The interface of Route Table does not exist in Interface Property.	Please create an Open Driver to add the missing interface or Contact NetBrain.
Interface Property	An interface mode is in (trunk, hybrid), and trunkNativeVlan is empty. OR the mode is not in (trunk, hybrid), and trunkNativeVlan is not empty.	Interface properties trunkNativeVlan and mode are inconsistent.	Please Contact NetBrain if the interface handles real traffic; otherwise, please ignore it.
Configuration and System Table	Device Configuration is more than one week older than the System Table.	Device Configuration is more than one week older than the System Table.	Please check if Configuration File is checked in Retrieve Live Data of Benchmark Tasks and Contact NetBrain if this issue still exists.
Overall Health Monitoring (OHM)	The values of OHM fields are not empty	Device data is missing in Overall Health View.	1. Please check the “CPU and Memory Usage” Parser in Parser Library. 2. Please check the “Overall Health View” Data View Template in the Data View Template Manager. 3. Contact NetBrain if this issue still exists.

3.2.3 Schedule PV

The system has one built-in Schedule PV task, which will be executed against all devices. The user cannot delete this task or add a new Schedule PV task. However, he can disable and enable this task.


Domain Management — Tenant: Initial Tenant — Domain: R10 Training — Operations — J.Zhang

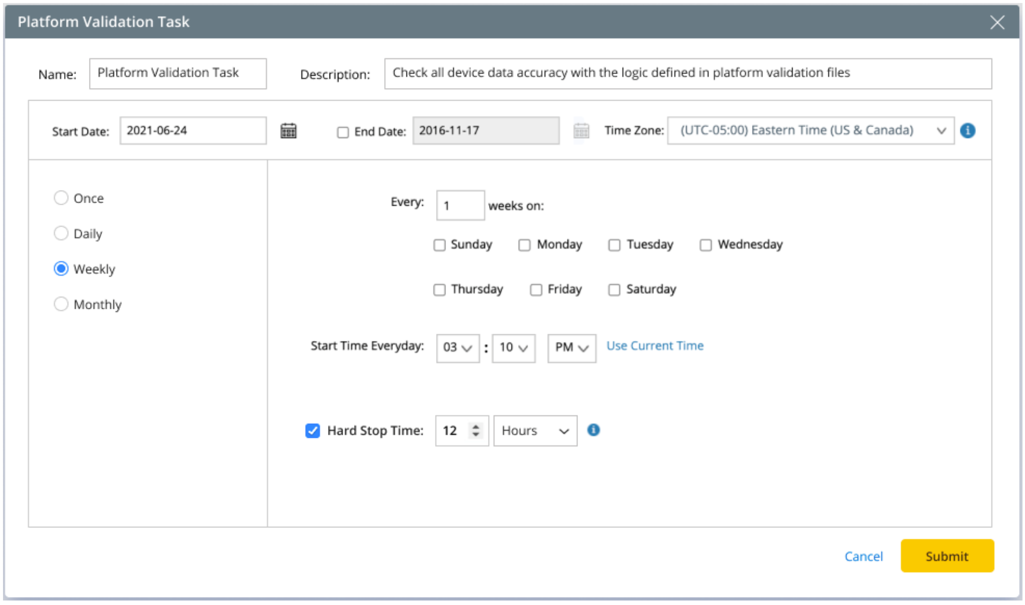
Start Page | Schedule Task

Schedule Discovery/Benchmark | Schedule Data View Template/Parser | Schedule Qapp | Schedule Plugin | Schedule Platform Validation

Refresh

Enable	Task Name	Last Run Time	Duration	Last Result	Current Status	Next Run Time	Frequency
✓	Platform Validation Check	6/7/2021, 12:11:42 PM	20 mins 10 secs	Succeeded	Idle	6/8/2021, 12:11:42 PM	Once

The user can set the schedule of the PV task, such as frequency, start date, end date, and start time. In addition, the system provides an optional setting, Hard Stop Time, which can be useful to stop the PV task after the specified hours so that it may not overlap with other scheduled tasks such as the system benchmark. The PV task will be paused with the current execution status recorded (primarily the devices to have been verified). Then, in the next PV cycle, PV tasks will be resumed from the last execution status (skip those devices which have been verified and verify the devices not verified in the last cycle yet.)


Platform Validation Task

Name: Platform Validation Task

Description: Check all device data accuracy with the logic defined in platform validation files

Start Date: 2021-06-24

☐ End Date: 2016-11-17

Time Zone: (UTC-05:00) Eastern Time (US & Canada)

○ Once
○ Daily
● Weekly
○ Monthly

Every: 1 weeks on:

☐ Sunday ☐ Monday ☐ Tuesday ☐ Wednesday

☐ Thursday ☐ Friday ☐ Saturday

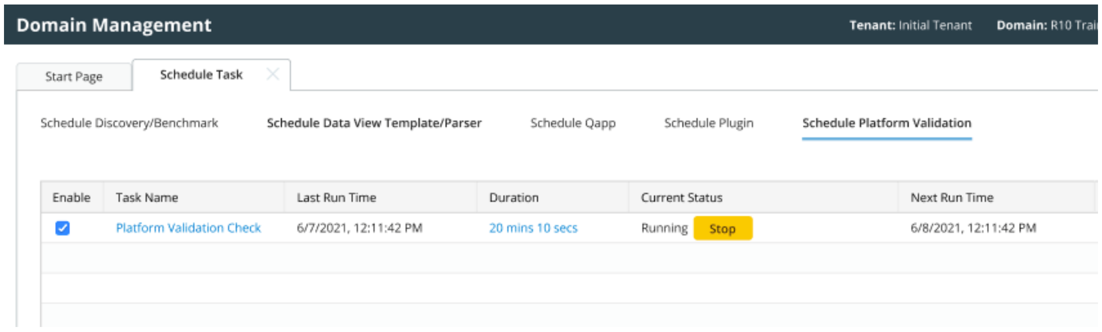
Start Time Everyday: 03 : 10 PM — Use Current Time

☑ Hard Stop Time: 12 Hours

Cancel | Submit

The PV task will start at the scheduled time or resume if the PV task of the last cycle does not finish within the hard stopped time. For each device in the domain, the PV task will run all PVRs to which this device is qualified (the qualification is defined in PV files) and enabled (a user can see all qualified PVRs and disable/enable a PVR on DAW).

The user can view PV status (Disabled, Running, Completed, Idle, and Paused), view the execution log, export the results to a CSV file, delete the results, and manually stop the running PV task.


Domain Management

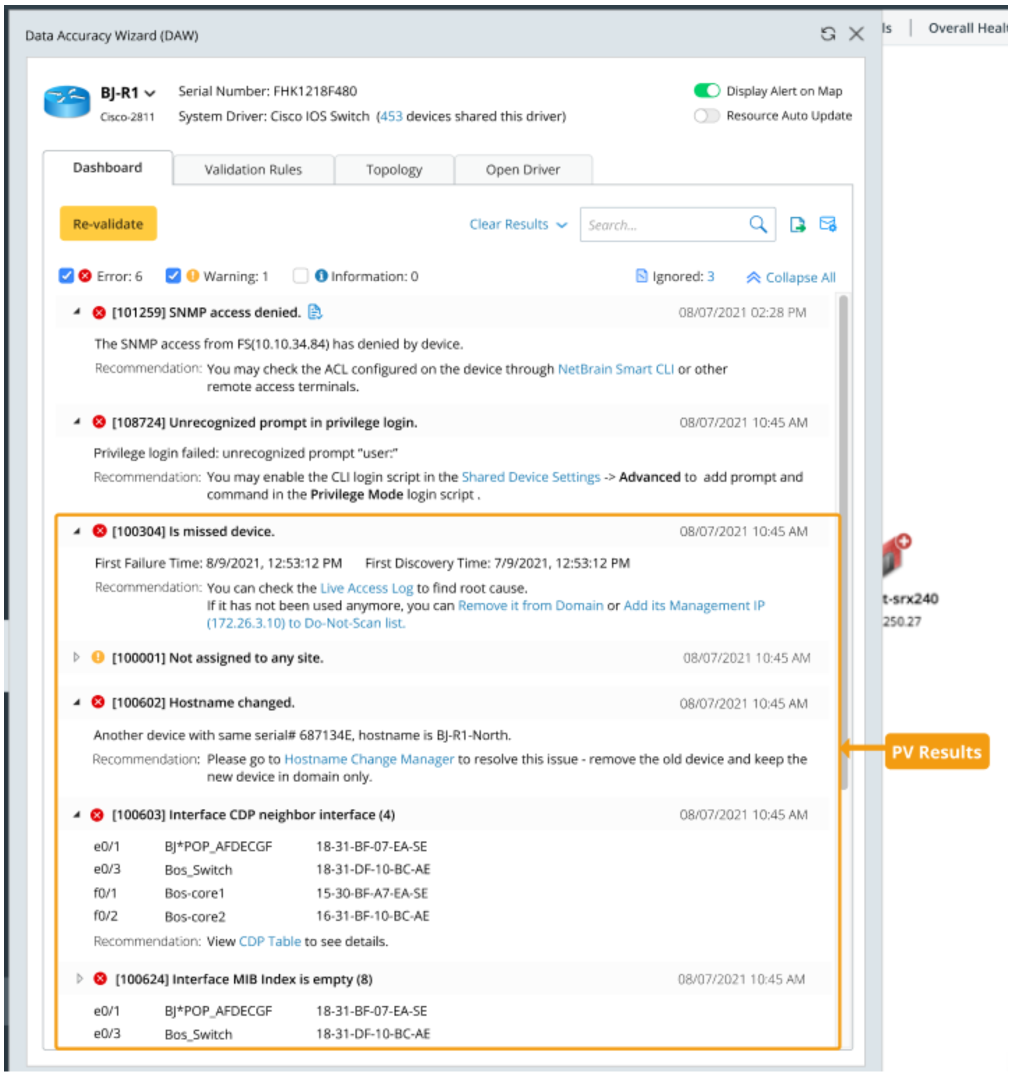
Tenant: Initial Tenant Domain: R10 Trai

Start Page
Schedule Task
Schedule Discovery/Benchmark
Schedule Data View Template/Parser
Schedule Qapp
Schedule Plugin
Schedule Platform Validation

Enable	Task Name	Last Run Time	Duration	Current Status	Next Run Time
✓	Platform Validation Check	6/7/2021, 12:11:42 PM	20 mins 10 secs	Running Stop	6/8/2021, 12:11:42 PM

3.2.4 View PV Results in DAW

The latest scheduled PV results are displayed on DAW Dashboard per device, categorized by **Error, Warning, and Information**. Each entry has the error code, the time of the first failure, error message, and recommendation. The link in the recommendation will bring the user the corresponding UI to fix the issue. After a user tries the recommended steps, he can click the button Re-validate to run all validation rules for this device to verify that the error is fixed.



Data Accuracy Wizard (DAW)

BJ-R1
Cisco-2811

Serial Number: FHK1218F480
System Driver: Cisco IOS Switch (453 devices shared this driver)

Display Alert on Map
Resource Auto Update

Dashboard | Validation Rules | Topology | Open Driver

Re-validate | Clear Results | Search...

Error: 6 | Warning: 1 | Information: 0 | Ignored: 3 | Collapse All

[101259] SNMP access denied. 08/07/2021 02:28 PM

The SNMP access from FS(10.10.34.84) has denied by device.

Recommendation: You may check the ACL configured on the device through NetBrain Smart CLI or other remote access terminals.

[108724] Unrecognized prompt in privilege login. 08/07/2021 10:45 AM

Privilege login failed: unrecognized prompt "user:"

Recommendation: You may enable the CLI login script in the Shared Device Settings -> **Advanced** to add prompt and command in the **Privilege Mode** login script .

[100304] Is missed device. 08/07/2021 10:45 AM

First Failure Time: 8/9/2021, 12:53:12 PM First Discovery Time: 7/9/2021, 12:53:12 PM

Recommendation: You can check the Live Access Log to find root cause.
If it has not been used anymore, you can Remove it from Domain or Add its Management IP (172.26.3.10) to Do-Not-Scan list.

[100001] Not assigned to any site. 08/07/2021 10:45 AM

[100602] Hostname changed. 08/07/2021 10:45 AM

Another device with same serial# 687134E, hostname is BJ-R1-North.

Recommendation: Please go to Hostname Change Manager to resolve this issue - remove the old device and keep the new device in domain only.

[100603] Interface CDP neighbor interface (4) 08/07/2021 10:45 AM

```
e0/1    BJ*POP_AFDECGF    18-31-BF-07-EA-SE
e0/3    Bos_Switch        18-31-DF-10-BC-AE
f0/1    Bos-core1         15-30-BF-A7-EA-SE
f0/2    Bos-core2         16-31-BF-10-BC-AE
```

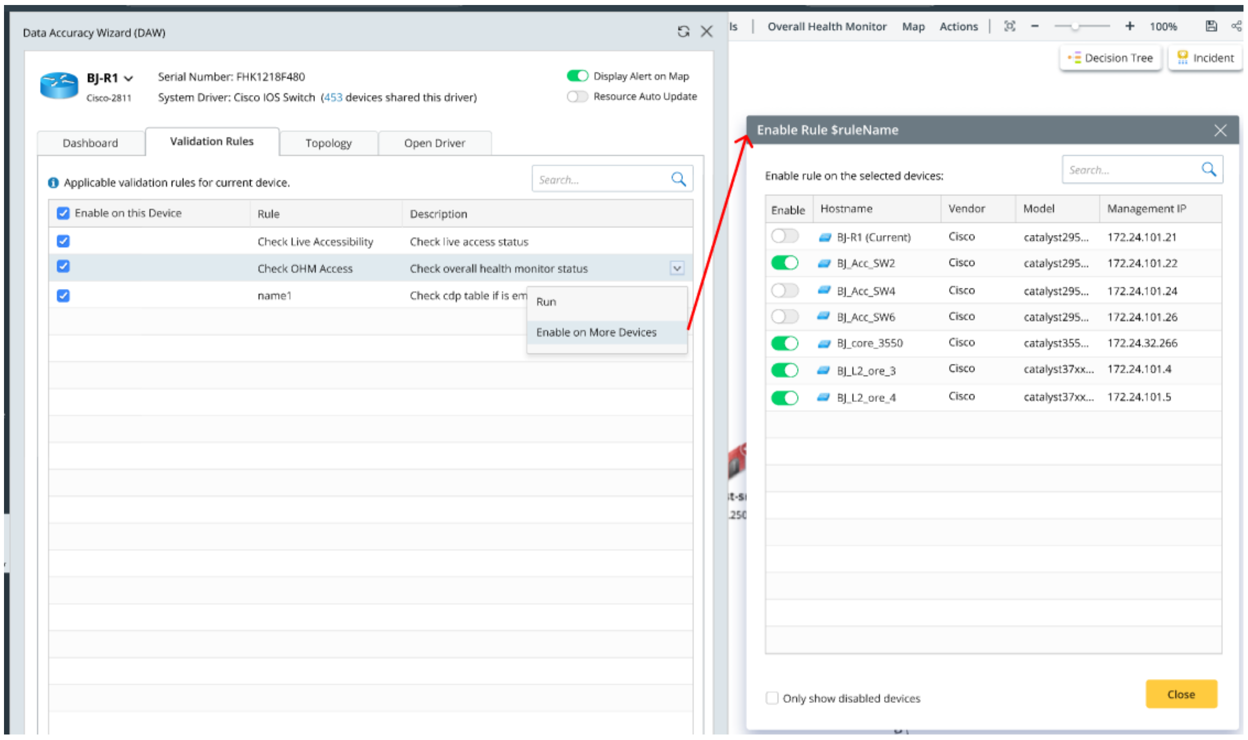
Recommendation: View CDP Table to see details.

[100624] Interface MIB Index is empty (8) 08/07/2021 10:45 AM

```
e0/1    BJ*POP_AFDECGF    18-31-BF-07-EA-SE
e0/3    Bos_Switch        18-31-DF-10-BC-AE
```

3.2.5 Enable, Disable and Run a PVR

All PVRs to which this device is qualified are listed under the Validation Rules tag of DAW. The user can disable and enable PVR for this device. The disabled PVR will not be run against this device in the scheduled PV task. The number of qualified devices is also displayed here for a user to see all devices qualified to a PVR and then disable or enable PVR for these devices.


Data Accuracy Wizard (DAW)

BJ-R1
Cisco-2811

Serial Number: FHK1218F480
System Driver: Cisco IOS Switch (453 devices shared this driver)

Display Alert on Map
Resource Auto Update

Applicable validation rules for current device.

☑ Enable on this Device	Rule	Description
☑	Check Live Accessibility	Check live access status
☑	Check OHM Access	Check overall health monitor status
☑	name1	Check cdp table if is em

Enable Rule $ruleName

Enable rule on the selected devices:

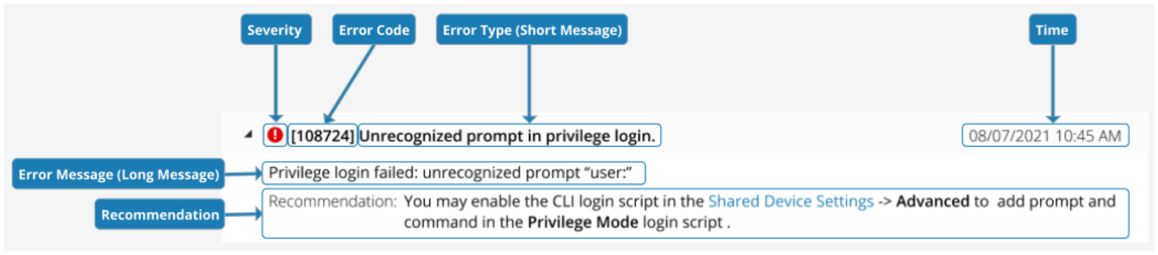
Enable	Hostname	Vendor	Model	Management IP
○	BJ-R1 (Current)	Cisco	catalyst295...	172.24.101.21
●	BJ_Acc_SW2	Cisco	catalyst295...	172.24.101.22
○	BJ_Acc_SW4	Cisco	catalyst295...	172.24.101.24
○	BJ_Acc_SW6	Cisco	catalyst295...	172.24.101.26
●	BJ_core_3550	Cisco	catalyst355...	172.24.32.266
●	BJ_L2_ore_3	Cisco	catalyst37xx...	172.24.101.4
●	BJ_L2_ore_4	Cisco	catalyst37xx...	172.24.101.5

☐ Only show disabled devices

3.3 System Validation (SV)

In V10.0 and earlier versions, the live access-related issues are scattered among many functions and UIs. As a result, the error messages are not consistent across these functions. IE v10.1 introduces System Validation (SV), which provides the system-wide consistent, more detailed, and accurate error messages in all features involving live access, such as discovery, benchmark, fine-tune, Qapp, NI, etc. These errors will be displayed in a central place, Data Accuracy Wizard (DAW). SV also summarizes the live access status on all devices and provides recommended solutions for each error type for users to maintain device data accuracy by themselves.

SV is a backend process requiring no user involvement. The output of SV is the system and live access-related error messages, which are displayed in the DAW dashboard. The SV error has the same components as PV error: error severity, unique error code, type, detailed message, and recommendations.



[108724] Unrecognized prompt in privilege login. 08/07/2021 10:45 AM

Privilege login failed: unrecognized prompt "user:"

Recommendation: You may enable the CLI login script in the Shared Device Settings -> **Advanced** to add prompt and command in the **Privilege Mode** login script .

3.3.1 SV Error Examples

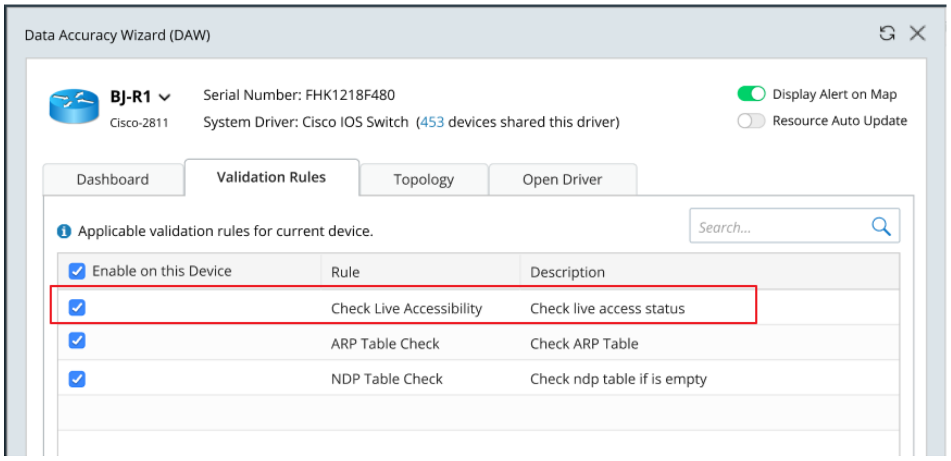
SV categorizes the live access failures into four major types: server error (Front Server, Front Server Group, Jumpbox, and API Server), Ping error, SNMP error, and CLI error (CLI connection, non-privileged login, privilege login, and advanced script). They could be further categorized into more detailed error types, and each has a corresponding recommended solution. The corresponding error message example and the recommended solution are listed below for each error type.

Category	Error Type	Error Message	Recommendation
Server Error	unavailable Front server	The front server FS (10.10.34.84) is unavailable.	You may switch an available front server in the Shared Device Settings or Tune Live Access. If there is no available front server, you can check the front server status in the **System Management** -> Front Server Controller. Contact your NetBrain admin if needed.
Ping Error	Unreachable IP address	The management IP address *10.10.34.84* of the device is unreachable.	You may switch an available management IP in the Shared Device Settings or Tune Live Access.
SNMP Error	SNMP access is declined.	The SNMP access from FS(10.10.34.84) has been declined by the device.	You may check the ACL configured on the device through NetBrain Smart CLI or other remote access terminals.

CLI Error	SSH/Telnet connect timed out.	SSH/Telnet connect from *FS(10.10.34.84)* failed: timed out	You may check the **CLI Connection Timeout** in the Shared Device Setting -> CLI -> Advanced.
CLI Error	Unrecognized prompt in non-privilege login.	Non-privilege login failed: unrecognized prompt *"user:"*	You may enable the CLI login script in the Shared Device Settings -> **Advanced** to add prompt and command in the **Non-privilege Mode** login script.
CLI Error	Authentication failed in privilege login.	Privilege login failed: invalid privilege username/password.	You may check the login credential configured in the Shared Device Setting or Tune Live Access.
CLI Error	Device not responding in *mode name*.	*Mode name login failed: device does not respond.*	You may check the **CLI Session Timeout** in the Shared Device Setting -> CLI -> Advanced.

3.3.2 Live Access Validation Rule

Even though live access failures could happen in all features involving the live access process, sometimes only partial live access will be checked in some tasks. For example, Ping and SNMP access won't be verified in the system benchmark. Therefore, live access validation rules are created to verify each live access method of devices to ensure the system can fully access them. Users can also re-run the validation process in the DAW dashboard to validate the live access settings (timeout settings, login script, etc.).



Serial Number: FHK1218F480
System Driver: Cisco IOS Switch (453 devices shared this driver)

Applicable validation rules for current device.

Enable on this Device	Rule	Description
✓	Check Live Accessibility	Check live access status
✓	ARP Table Check	Check ARP Table
✓	NDP Table Check	Check ndp table if is empty

Live Access Validate Rule will validate the accessibility of devices based on the latest device settings with the following steps:

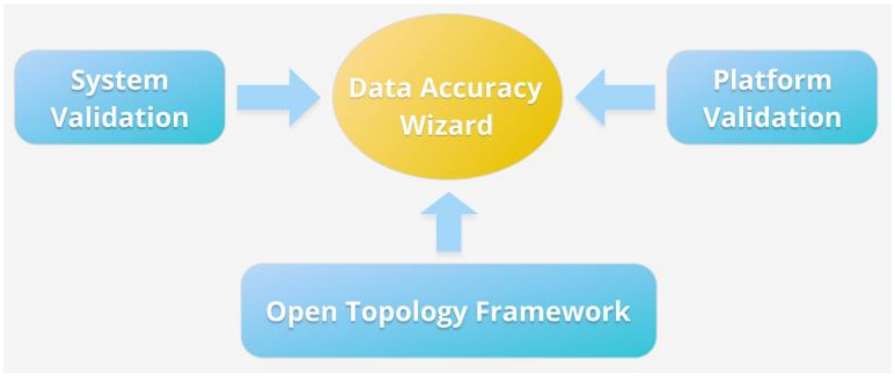
- Check the connection of relevant servers, including the front server, front server group, jumpbox server, and API server.
- Ping, the device management IP, to validate the connection.
- Access the device via SNMP using the latest SNMP session timeout setting.
- Validate the CLI access of the device using the latest CLI session timeout setting. The CLI access includes SSH/Telnet connection and authentication in non-privilege mode, privilege mode, and customized mode.

If live access failure happens in any of the steps above, corresponding error messages and codes will be generated.

3.4 Data Accuracy Wizard (DAW)

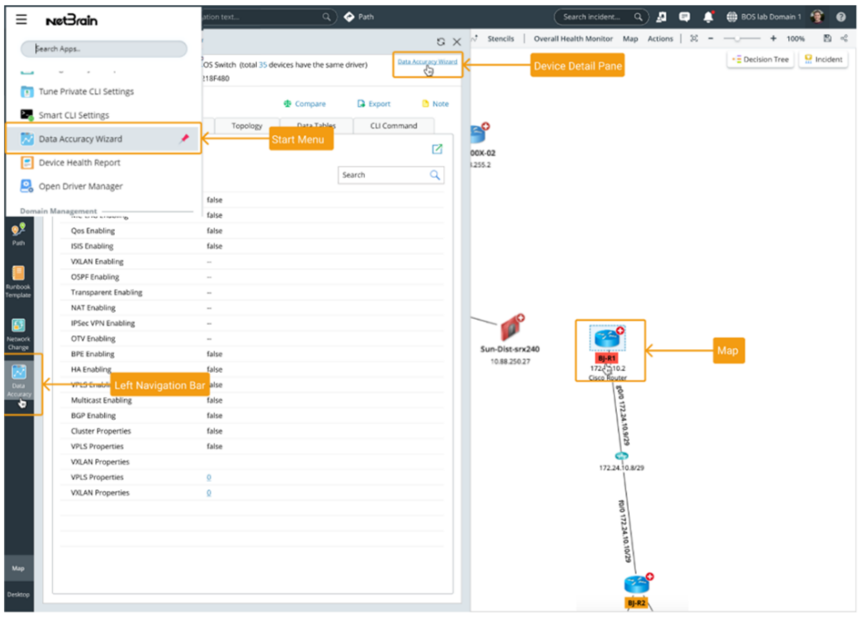
The data accuracy issues can be caused by many functions and misconfigurations. The system can detect these issues and provide the functions to fix these issues. However, these issues are scattered in many places, such as benchmark reports, domain health reports, tune live access, duplicated IP manager, etc. The system does not have a central place to list all data accuracy issues and the methods to fix these issues.

V10.1 introduces Data Accuracy Wizard (DAW), a feature dedicated to all data accuracy issues per device, such as live access issues from SV, the device data issue from PV, and the topology results from OT. The verification rules and the open drivers are also listed in DAW so that a user can enable/disable these rules and drivers.



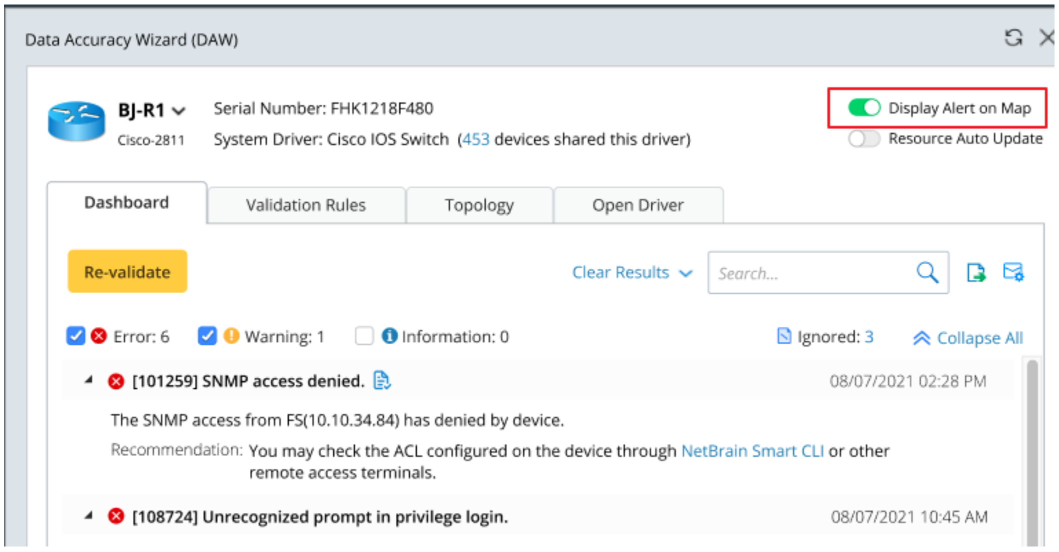
3.4.1 Open DAW

If the devices on the map have SV/PV alerts, the corresponding hostnames will be highlighted with an alert icon. Users can click the icon to open DAW for this device. DAW can be opened from the Start menu, left navigation bar, and device detail pane.



	false
	false
Qos Enabling	false
ISIS Enabling	false
VXLAN Enabling	--
OSPF Enabling	--
Transparent Enabling	--
NAT Enabling	--
IPSec VPN Enabling	--
OTV Enabling	--
BPE Enabling	false
HA Enabling	false
VPLS Enabl	alse
Multicast Enabling	false
BGP Enabling	false
Cluster Properties	false
VPLS Properties	false
VXLAN Properties	
VPLS Properties	0
VXLAN Properties	0

Users can enable or disable DAW alert in the map with the option **Display Alert on Map**.



Serial Number: FHK1218F480

System Driver: Cisco IOS Switch (453 devices shared this driver)

[101259] SNMP access denied. 08/07/2021 02:28 PM

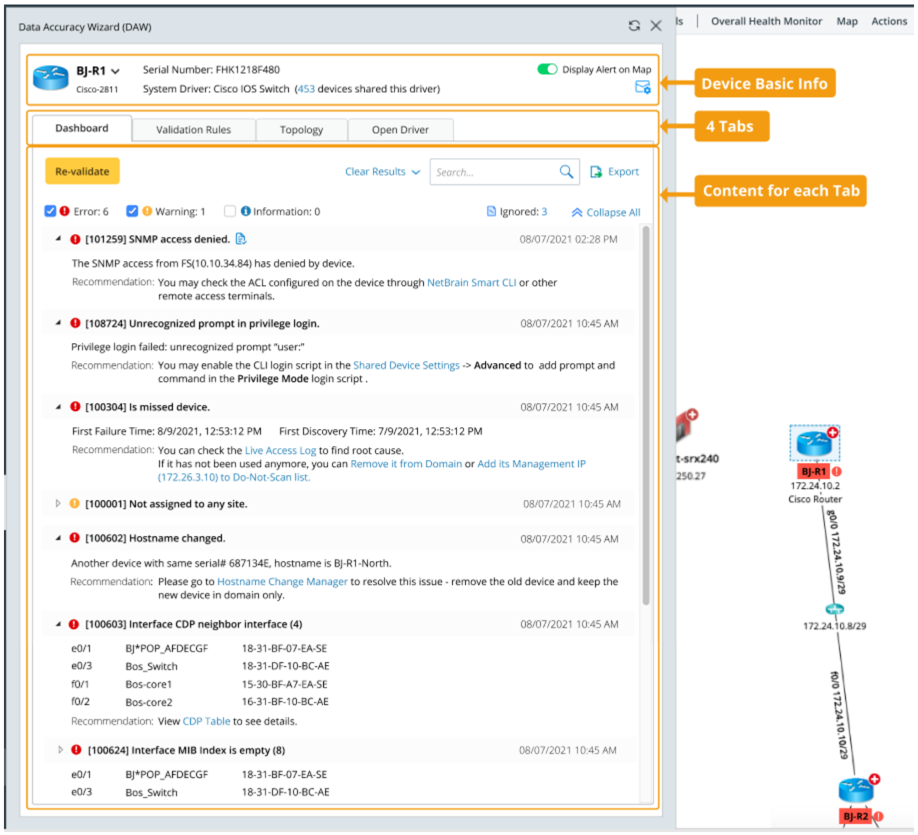
The SNMP access from FS(10.10.34.84) has denied by device.

Recommendation: You may check the ACL configured on the device through NetBrain Smart CLI or other remote access terminals.

[108724] Unrecognized prompt in privilege login. 08/07/2021 10:45 AM

3.4.2 DAW Components

DAW is for one device only, and the user can change the device. At the top of DAW, the hostname and the basic data are displayed.



BJ-R1
Cisco-2811
Serial Number: FHK1218F480
System Driver: Cisco IOS Switch (453 devices shared this driver)

Error: 6 Warning: 1 Information: 0 Ignored: 3

[101259] SNMP access denied. 08/07/2021 02:28 PM
The SNMP access from FS(10.10.34.84) has denied by device.
Recommendation: You may check the ACL configured on the device through NetBrain Smart CLI or other remote access terminals.

[108724] Unrecognized prompt in privilege login. 08/07/2021 10:45 AM
Privilege login failed: unrecognized prompt "user:"
Recommendation: You may enable the CLI login script in the Shared Device Settings -> **Advanced** to add prompt and command in the **Privilege Mode** login script .

[100304] Is missed device. 08/07/2021 10:45 AM
First Failure Time: 8/9/2021, 12:53:12 PM First Discovery Time: 7/9/2021, 12:53:12 PM
Recommendation: You can check the Live Access Log to find root cause.
If it has not been used anymore, you can Remove it from Domain or Add its Management IP (172.26.3.10) to Do-Not-Scan list.

[100001] Not assigned to any site. 08/07/2021 10:45 AM

[100602] Hostname changed. 08/07/2021 10:45 AM
Another device with same serial# 687134E, hostname is BJ-R1-North.
Recommendation: Please go to Hostname Change Manager to resolve this issue - remove the old device and keep the new device in domain only.

[100603] Interface CDP neighbor interface (4) 08/07/2021 10:45 AM

e0/1	BJ*POP_AFDECGF	18-31-BF-07-EA-SE
e0/3	Bos_Switch	18-31-DF-10-BC-AE
f0/1	Bos-core1	15-30-BF-A7-EA-SE
f0/2	Bos-core2	16-31-BF-10-BC-AE

Recommendation: View CDP Table to see details.

[100624] Interface MIB Index is empty (8) 08/07/2021 10:45 AM

e0/1	BJ*POP_AFDECGF	18-31-BF-07-EA-SE
e0/3	Bos_Switch	18-31-DF-10-BC-AE

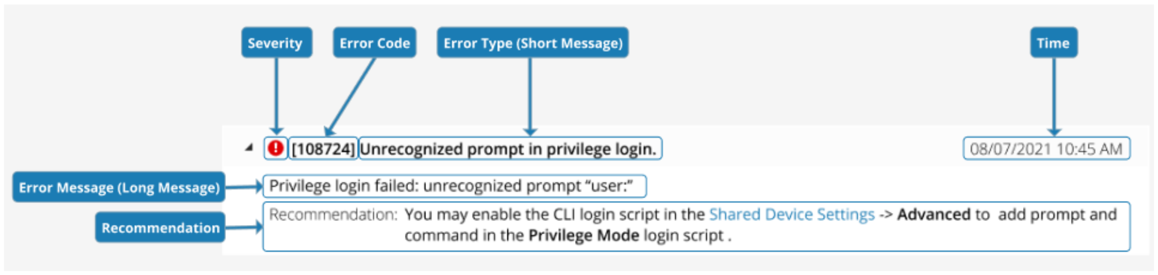
DAW UI has four tabs:

- **Dashboard**: display all data issues created by SV and PV. Each issue has the error code, error message, and recommended actions.
- **Validation Rules**: display all PVRs applicable for this device. The user can enable, disable or debug run a PVR. Refer to section 3.2 for details.
- **Topology**: displays OT results, including VLAN groups to which this device belongs, L3 and L2 topology links. Refer to section 3.1 for details.
- **Open Driver**: displays all ODs applicable to this device. The user can enable, disable and test run an OD. Refer to section 3.5 for details.

3.4.3 DAW Dashboard

The issues reported by SV and PV are listed under the Dashboard tag. SV is a backend process to collect all system and live access issues per device.

The error message will be displayed as below:



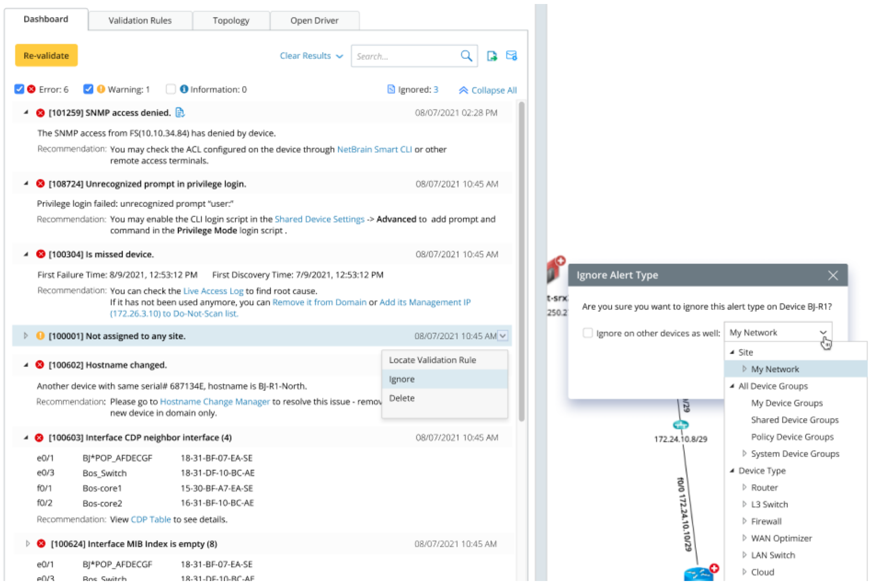
[108724] Unrecognized prompt in privilege login. 08/07/2021 10:45 AM

Privilege login failed: unrecognized prompt "user:"

Recommendation: You may enable the CLI login script in the Shared Device Settings -> **Advanced** to add prompt and command in the **Privilege Mode** login script .

For each error, the users can:

- Use the link provided in the recommendation to fix the issue, and then click the re-validate button on the DAW dashboard to confirm that the issue has been fixed.
- Ignore the error from the right-click menu to remove it from the dashboard. The user can ignore an alert type on a device, all devices, or a group of devices. An ignored error can be restored.



[101259] SNMP access denied. 08/07/2021 02:28 PM

The SNMP access from FS(10.10.34.84) has denied by device.

Recommendation: You may check the ACL configured on the device through NetBrain Smart CLI or other remote access terminals.

[108724] Unrecognized prompt in privilege login. 08/07/2021 10:45 AM

Privilege login failed: unrecognized prompt "user:"

Recommendation: You may enable the CLI login script in the Shared Device Settings -> **Advanced** to add prompt and command in the **Privilege Mode** login script .

[100304] Is missed device. 08/07/2021 10:45 AM

First Failure Time: 8/9/2021, 12:53:12 PM First Discovery Time: 7/9/2021, 12:53:12 PM

Recommendation: You can check the Live Access Log to find root cause. If it has not been used anymore, you can Remove it from Domain or Add its Management IP (172.26.3.10) to Do-Not-Scan list.

[100001] Not assigned to any site. 08/07/2021 10:45 AM

[100602] Hostname changed.

Another device with same serial# 687134E, hostname is BJ-R1-North.

Recommendation: Please go to Hostname Change Manager to resolve this issue - remov new device in domain only.

[100603] Interface CDP neighbor interface (4) 08/07/2021 10:45 AM

e0/1	BJ*POP_AFDECGF	18-31-BF-07-EA-SE
e0/3	Bos_Switch	18-31-DF-10-BC-AE
f0/1	Bos-core1	15-30-BF-A7-EA-SE
f0/2	Bos-core2	16-31-BF-10-BC-AE

Recommendation: View CDP Table to see details.

[100624] Interface MIB Index is empty (8) 08/07/2021 10:45 AM

e0/1	BJ*POP_AFDECGF	18-31-BF-07-EA-SE
e0/3	Bos_Switch	18-31-DF-10-BC-AE

Ignore Alert Type

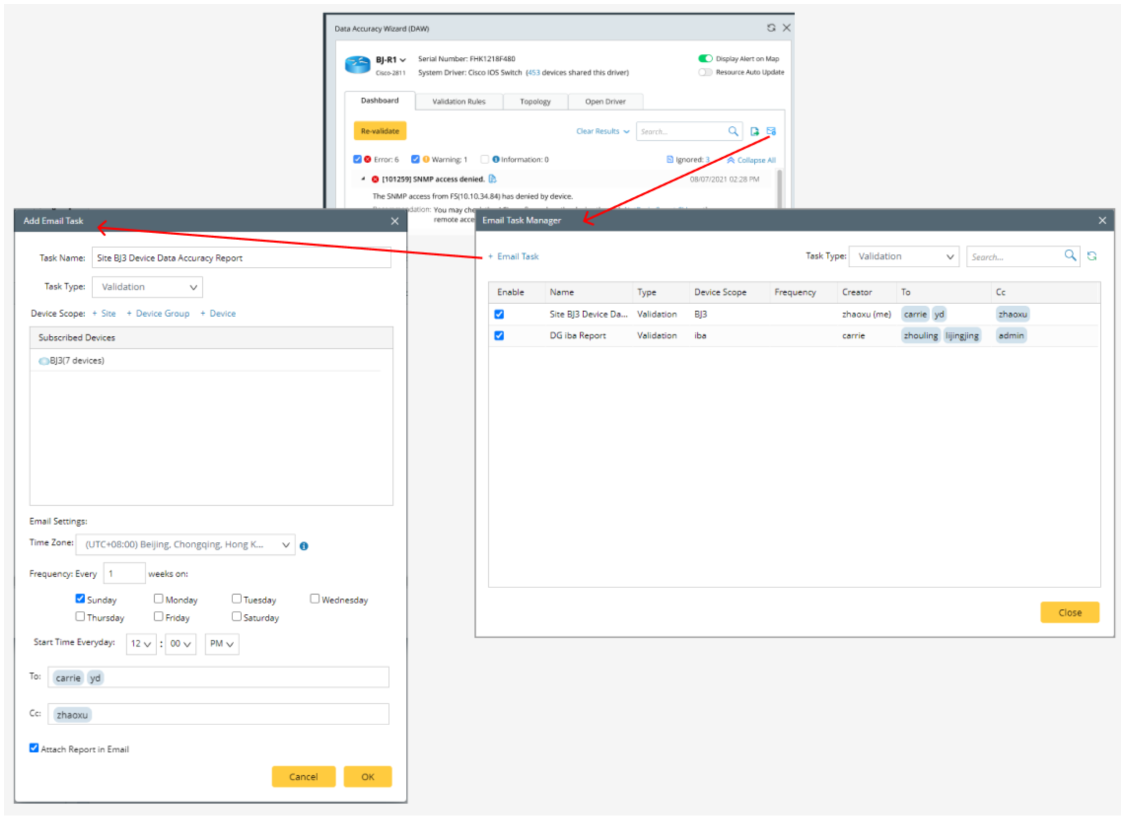
Are you sure you want to ignore this alert type on Device BJ-R1?

Ignore on other devices as well: My Network

- Locate the Validation Rule and disable the rule if users think that the rule is not relevant. The user can enable or disable the rule in one device for all qualified devices.

3.4.4 Notification of DAW reports

Users can set up an email task to receive the DAW report of the devices.



Serial Number: FHK1218F480
System Driver: Cisco IOS Switch (453 devices shared this driver)

[101259] SNMP access denied. 08/07/2021 02:28 PM
The SNMP access from FS(10.10.34.84) has denied by device.

Add Email Task

Task Name: Site BJ3 Device Data Accuracy Report
Task Type: Validation
Device Scope: + Site + Device Group + Device

Subscribed Devices
BJ3(7 devices)

Email Settings:
Time Zone: (UTC+08:00) Beijing, Chongqing, Hong K...
Frequency: Every 1 weeks on:
[✓ Sunday] Monday Tuesday Wednesday Thursday Friday Saturday
Start Time Everyday: 12 : 00 PM
To: carrie yd
Cc: zhaoxu
[✓ Attach Report in Email]

Cancel OK

Email Task Manager

+ Email Task
Task Type: Validation

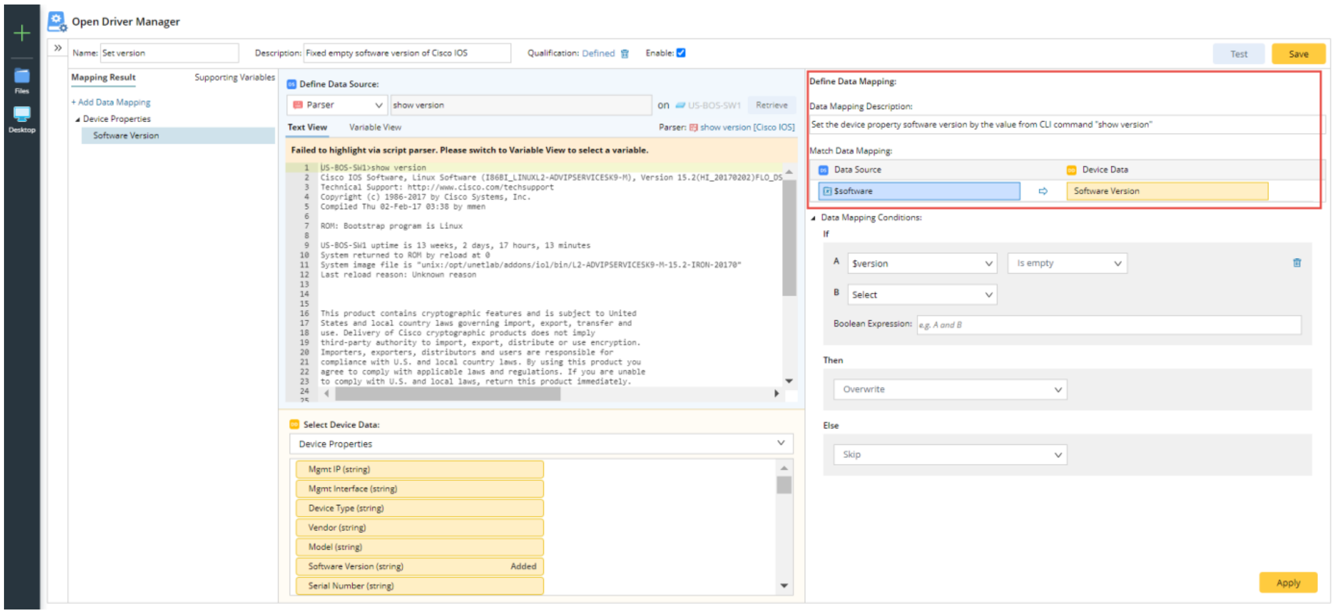
Enable	Name	Type	Device Scope	Frequency	Creator	To	Cc
✓	Site BJ3 Device Da...	Validation	BJ3		zhaoxu (me)	carrie yd	zhaoxu
✓	DG iba Report	Validation	iba		carrie	zhouling lijingjing	admin

Close

3.5 Open Driver

In IE v10.0 and earlier versions, the device data are retrieved and parsed by the device driver (System Driver), maintained, and updated by NetBrain. When device data is inaccurate, the user can only contact NetBrain and wait for a patch of the driver to fix the issue. V10.1 enables users to correct the data accuracy issues by themselves via a new feature, Open Driver (OD), which leverages the low code ability of Visual Parser.

An Open Driver defines the target device data (which data is to be corrected), the data source (the commands to retrieve the data), the Parser, the mapping between parser variables and the target data. For example, suppose the software version of a Cisco IOS router is empty or incorrect. The user can create an Open Driver to parse the output of the CLI command ***show version*** and map the variable ***$version*** to the device property ***Software Version***.



Name: Set version

Description: Fixed empty software version of Cisco IOS

Qualification: Defined

Enable: ✓

Define Data Source:

Parser: show version on US-BOS-SW1

Parser: show version [Cisco IOS]

Failed to highlight via script parser. Please switch to Variable View to select a variable.

```
US-BOS-SW1>show version
Cisco IOS Software, Linux Software (I86BI_LINUXL2-ADVIPSERVICESK9-M), Version 15.2(HI_20170202)FLO_DS
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2017 by Cisco Systems, Inc.
Compiled Thu 02-Feb-17 03:38 by mmen

ROM: Bootstrap program is Linux

US-BOS-SW1 uptime is 13 weeks, 2 days, 17 hours, 13 minutes
System returned to ROM by reload at 0
System image file is "unix:/opt/unetlab/addons/iol/bin/L2-ADVIPSERVICESK9-M-15.2-IRON-20170"
Last reload reason: Unknown reason



This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.
```

Select Device Data: Device Properties

- Mgmt IP (string)
- Mgmt Interface (string)
- Device Type (string)
- Vendor (string)
- Model (string)
- Software Version (string) Added
- Serial Number (string)

Define Data Mapping:

Data Mapping Description: Set the device property software version by the value from CLI command "show version"

Match Data Mapping:

Data Source		Device Data
$software	⇨	Software Version

Data Mapping Conditions:

If

A: $version — Is empty

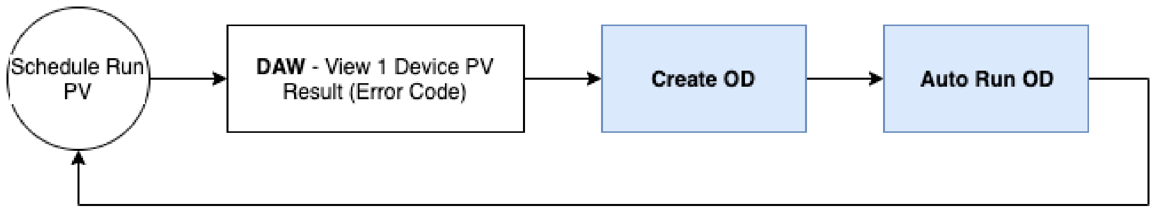
B: Select

Boolean Expression: e.g. A and B

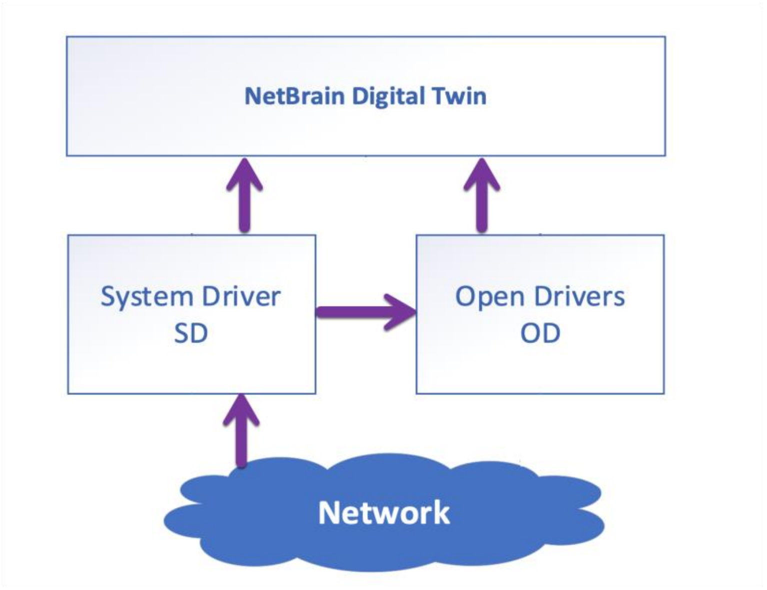
Then: Overwrite

Else: Skip

The main workflow: after the scheduled PV, NetBrain users view PV errors and warnings on DAW. For the issues recommended to be resolved by OD, create an OD and test run OD.



A user can define an OD to modify multiple target data (for example, the same CLI command can set more than one device property) and define as many ODs as necessary for one device. ODs will be executed automatically after the SD whenever the corresponding SD is applied, such as the Benchmark process.



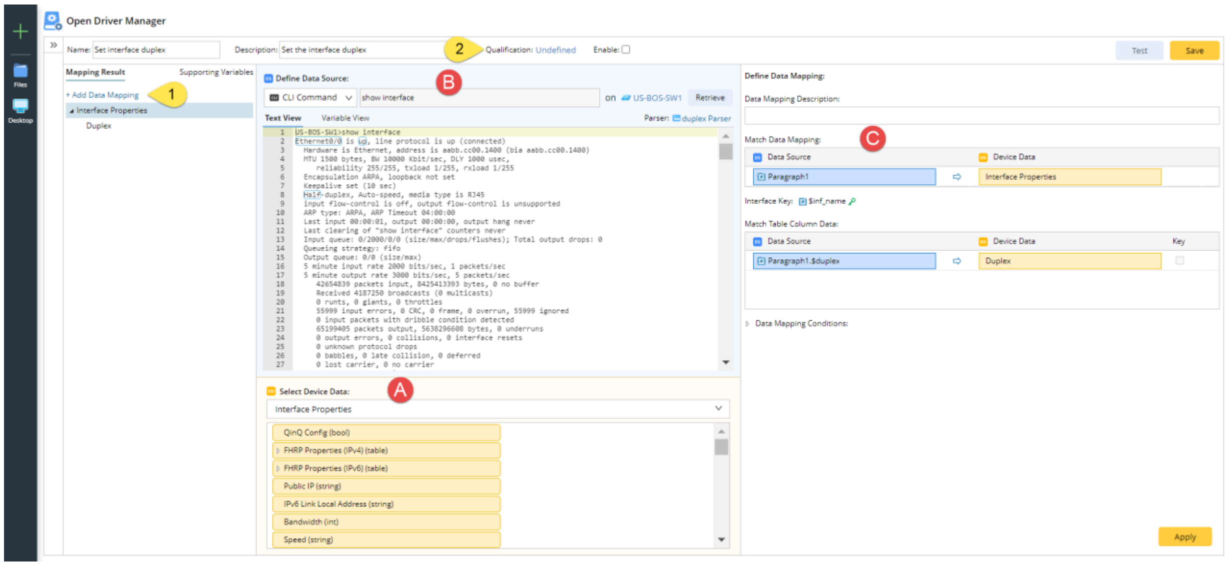
3.5.1 Define OD

The system provides the OD editor to define OD without coding. The user can follow the recommended action, **New OD/Enable Beta OD** at DAW, or from **Open Driver Manager** to create an OD.

The main steps to define an OD is to define the data mapping, which includes three essential steps:

1. Select the target device data. You can select a device property, an interface property, a system table, or NCT.
2. Define the data source. You can use the variables from an existing Parser or create a new Parser.
3. Map the source data variables to the target device data.

After that, you can define the qualifications for this OD.



```
US-BOS-SW1>show interface
Ethernet0/0 is up, line protocol is up (connected)
  Hardware is Ethernet, address is aabb.cc00.1400 (bia aabb.cc00.1400)
  MTU 1500 bytes, BW 10000 Kbit/sec, DLY 1000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Half-duplex, Auto-speed, media type is RJ45
  input flow-control is off, output flow-control is unsupported
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:01, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/2000/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 2000 bits/sec, 1 packets/sec
  5 minute output rate 3000 bits/sec, 5 packets/sec
     42654839 packets input, 8425413393 bytes, 0 no buffer
     Received 4187250 broadcasts (0 multicasts)
     0 runts, 0 giants, 0 throttles
     55999 input errors, 0 CRC, 0 frame, 0 overrun, 55999 ignored
     0 input packets with dribble condition detected
     65199405 packets output, 5638296608 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 unknown protocol drops
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
```

3.5.1.1 Select Target Device Data

OD can be used to modify the following data:

- **Device Property (GDR)**

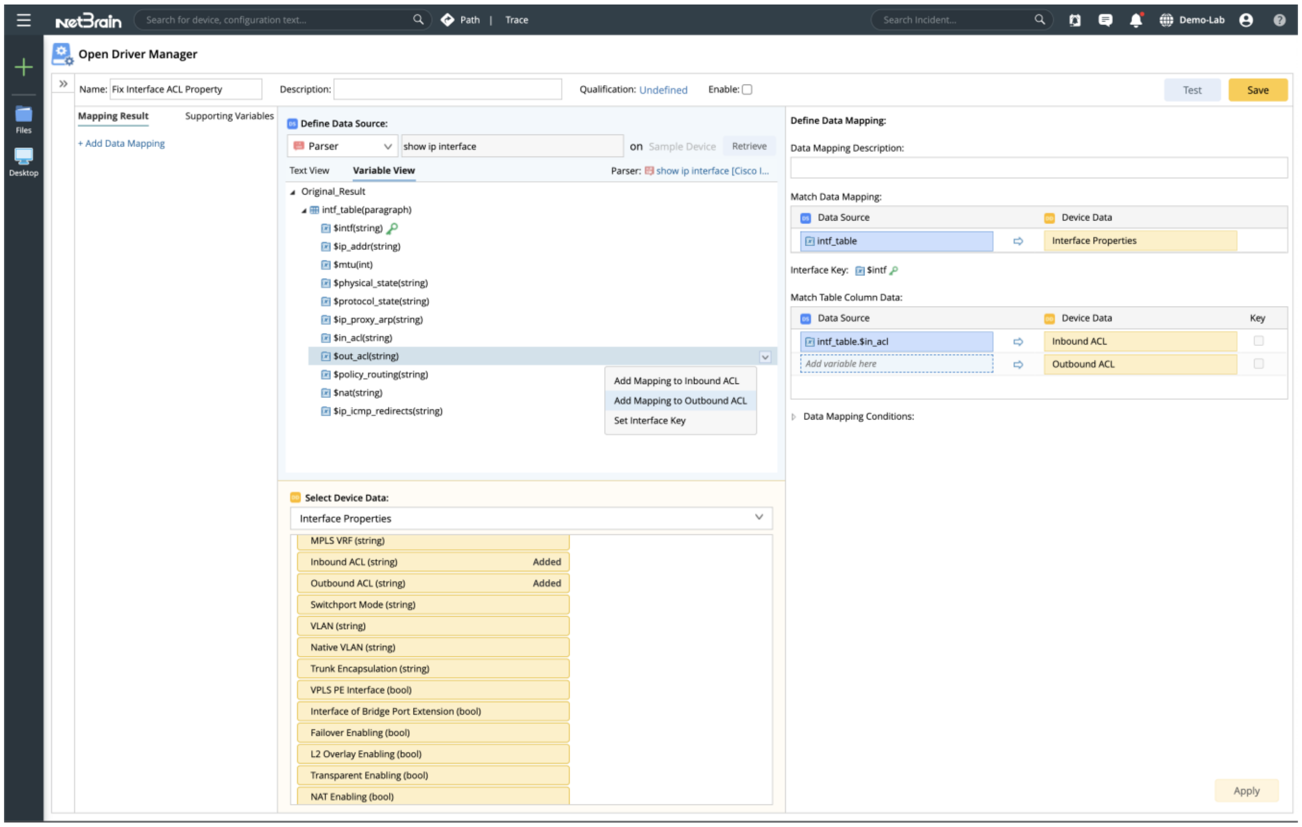
Not all GDR can be modified by OD. Only those with the flag Visible in OD checked (can be set in the GDR configuration page at Tenant Management) can be changed by OD. The system supports the following types of device GDR:

	Description
Single Value (String, integer, Boolean)	For example, Hostname, Software Version, etc.
Table	Can select any column of the table as the target data.
List	Can be mapped to a column (field) of a source table.
Object	Support the property of an object.

- **Interface Property (GDR)**

Same as the device GDR, only those interface GDR with the flag Visible in OD checked can be modified by OD.

Interface GDR is a table with the interface name as the key from a device's perspective. Therefore, the source data must be a table with the interface name as the key. For example, you can use the CLI command, ***show ip interface***, to set the interface, inbound ACL, outbound ACL, etc.



Name: Fix Interface ACL Property
Description:
Qualification: Undefined
Enable: ☐

Define Data Source:
Parser | show ip interface | on Sample Device | Retrieve
Parser: show ip interface [Cisco I...

- Original_Result
  - intf_table(paragraph)
    - $intf(string)
    - $ip_addr(string)
    - $mtu(int)
    - $physical_state(string)
    - $protocol_state(string)
    - $ip_proxy_arp(string)
    - $in_acl(string)
    - $out_acl(string)
    - $policy_routing(string)
    - $nat(string)
    - $ip_icmp_redirects(string)

Select Device Data:
Interface Properties

MPLS VRF (string)	
Inbound ACL (string)	Added
Outbound ACL (string)	Added
Switchport Mode (string)	
VLAN (string)	
Native VLAN (string)	
Trunk Encapsulation (string)	
VPLS PE Interface (bool)	
Interface of Bridge Port Extension (bool)	
Failover Enabling (bool)	
L2 Overlay Enabling (bool)	
Transparent Enabling (bool)	
NAT Enabling (bool)	

Define Data Mapping:
Data Mapping Description:

Match Data Mapping:

Data Source	Device Data
intf_table	Interface Properties

Interface Key: $intf

Match Table Column Data:

Data Source	Device Data	Key
intf_table.$in_acl	Inbound ACL	☐
Add variable here	Outbound ACL	☐

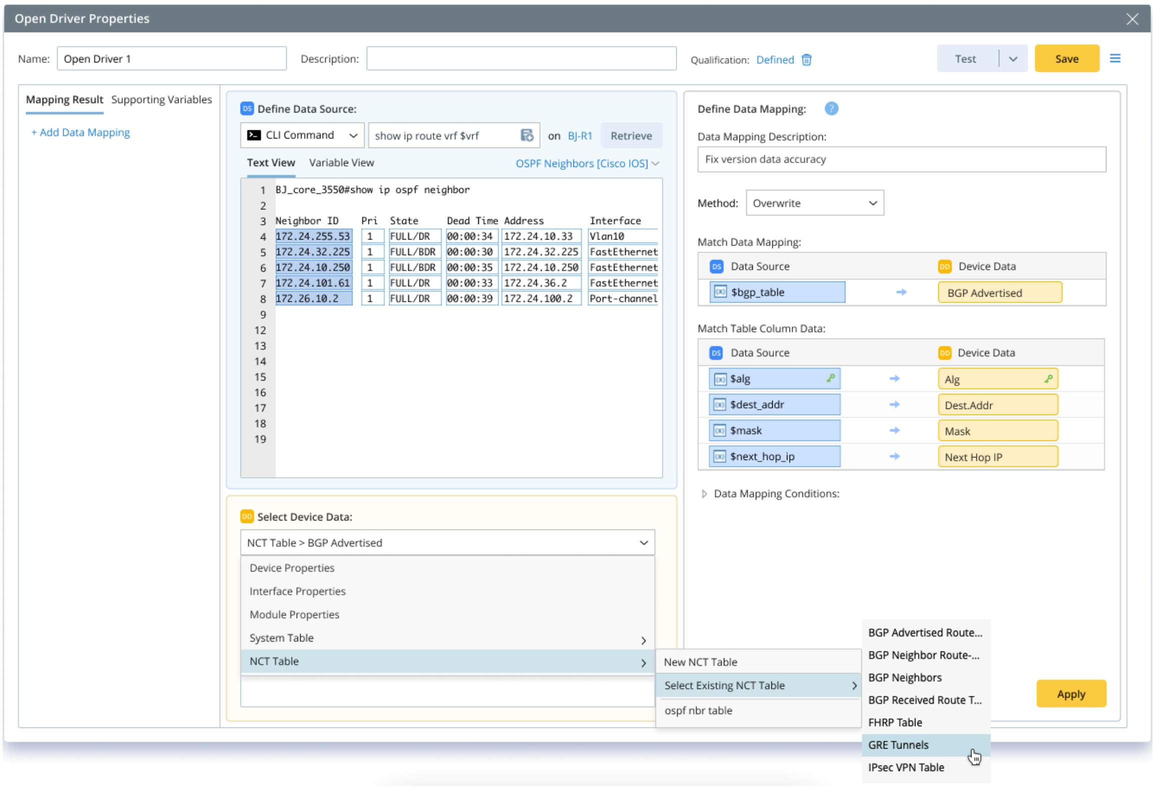
The system supports the same types of interface GDR as device GDR.

- **System Table**

Built-in system tables are created by SD and can be modified by OD.

- **NCT Table**

OD can modify an existing NCT table and create a new NCT table. Only the global NCT table (not VRF or subtables) is supported. OD can change all columns of an existing or new NCT table.



Name: Open Driver 1

Description:

Qualification: Defined

CLI Command: show ip route vrf $vrf on BJ-R1

```
BJ_core_3550#show ip ospf neighbor

Neighbor ID     Pri  State      Dead Time Address        Interface
172.24.255.53   1    FULL/DR    00:00:34  172.24.10.33   Vlan10
172.24.32.225   1    FULL/BDR   00:00:30  172.24.32.225  FastEthernet
172.24.10.250   1    FULL/BDR   00:00:35  172.24.10.250  FastEthernet
172.24.101.61   1    FULL/DR    00:00:33  172.24.36.2    FastEthernet
172.26.10.2     1    FULL/DR    00:00:39  172.24.100.2   Port-channel
```

NCT Table > BGP Advertised

Data Mapping Description: Fix version data accuracy

Method: Overwrite

Match Data Mapping:

Data Source	Device Data
$bgp_table	BGP Advertised

Match Table Column Data:

Data Source	Device Data
$alg	Alg
$dest_addr	Dest.Addr
$mask	Mask
$next_hop_ip	Next Hop IP

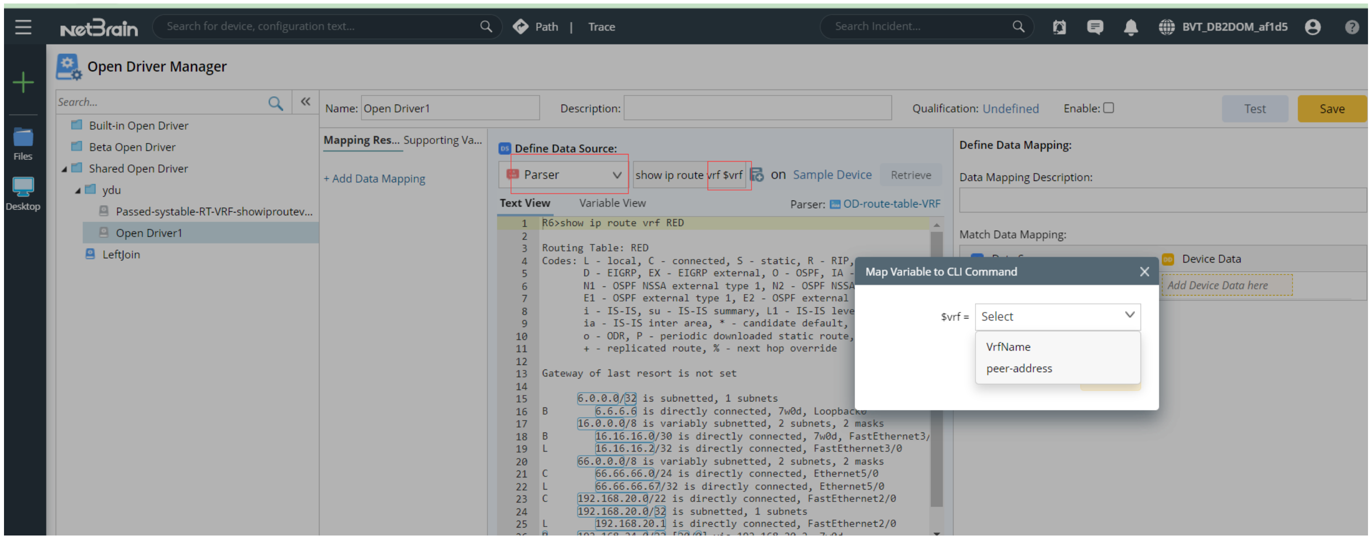
The NCT table created by OD is identical to the standard NCT table, can be updated in the system benchmark process, is viewed as device data, and used by the automation assets such as Qapp, Runbook nods, Qapp, and Change Analysis.

3.5.1.2 Define Data Source

The data source of an OD can be a CLI command, configuration, and an existing Parser. To support the data retrieved via SNMP or API as the data source, you need to define a Parser for the SNMP or API data first and then select the Parser as the data source.

To add a CLI command as the data source, the user enters the CLI command, retrieves the data, and selects an existing Parser or creates a new Parser. Users can choose a Parser with the CLI Command having the parameters such as ***show ip route vrf $VrfName***(however, users cannot enter the CLI command with the parameters directly). V10.1 only supports two built-in variables:

Built-in Variable	Example
$VrfName	show ip route vrf $vrf_name
$peer-address	show ip bgp advertised $peer_address show ip bgp advertised vrf $peer_address $vrf_name


Open Driver Manager

Name: Open Driver1 Description: Qualification: Undefined Enable: ☐

Define Data Source: Parser | show ip route vrf $vrf on Sample Device

Parser: OD-route-table-VRF

```
R6>show ip route vrf RED

Routing Table: RED
Codes: L - local, C - connected, S - static, R - RIP,
       D - EIGRP, EX - EIGRP external, O - OSPF, IA -
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA
       E1 - OSPF external type 1, E2 - OSPF external
       i - IS-IS, su - IS-IS summary, L1 - IS-IS leve
       ia - IS-IS inter area, * - candidate default,
       o - ODR, P - periodic downloaded static route,
       + - replicated route, % - next hop override

Gateway of last resort is not set

      6.0.0.0/32 is subnetted, 1 subnets
B        6.6.6.6 is directly connected, 7w0d, Loopback0
      16.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
B        16.16.16.0/30 is directly connected, 7w0d, FastEthernet3/
L        16.16.16.2/32 is directly connected, FastEthernet3/0
      66.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        66.66.66.0/24 is directly connected, Ethernet5/0
L        66.66.66.67/32 is directly connected, Ethernet5/0
C     192.168.20.0/22 is directly connected, FastEthernet2/0
      192.168.20.0/32 is subnetted, 1 subnets
L        192.168.20.1 is directly connected, FastEthernet2/0
```

Map Variable to CLI Command: $vrf = Select (VrfName, peer-address)

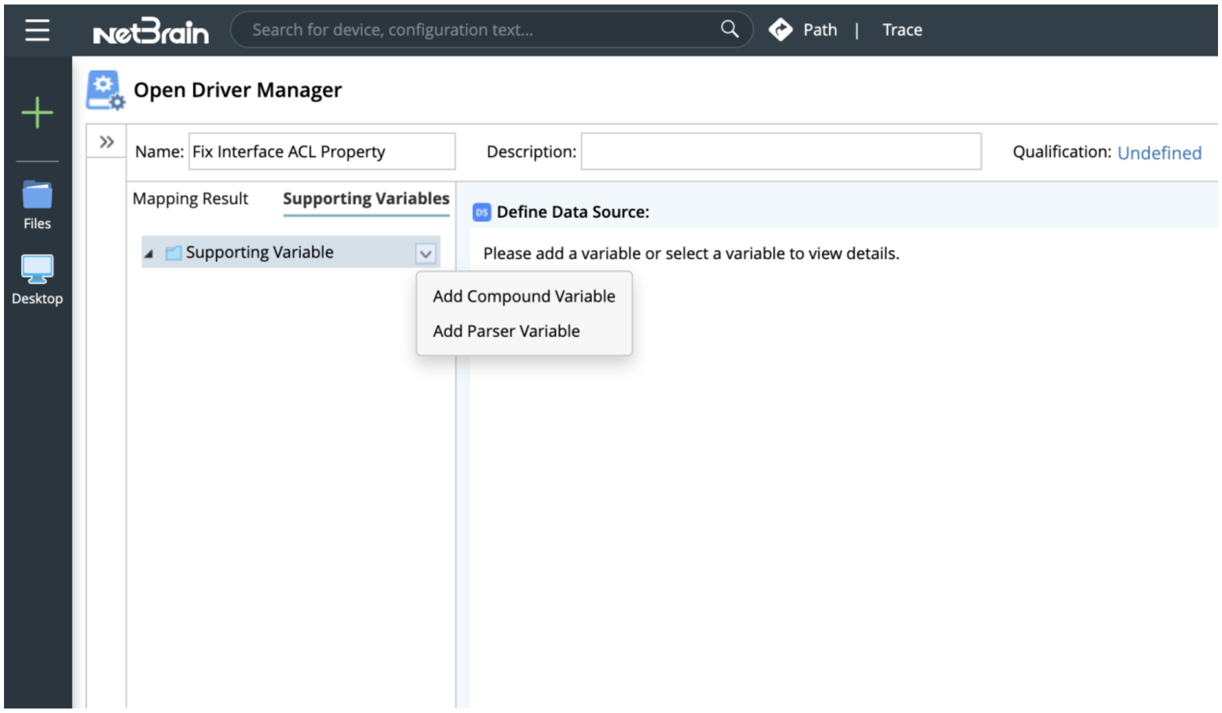
The configuration is a special data source. The user can retrieve the data from the live network or use the cached data and then select a Parser or create a new Parser.

All variables defined by the data source Parser can be mapped to the target device data. However, in some cases, the variables of a Parser cannot be directly mapped to a target device data. For example, the CLI command parser variable is an IP address, while the target device data is the device hostname. Or the variables obtained from two CLI Parsers need to be merged to a new variable to be mapped to a target device data.

To support these use cases, which require secondary parser data processing, the user will define the supporting variables first and then map the variable to the target device data. A support variable can be any Parser variable or a compound variable.

A compound variable can combine multiple Parser variables into a new variable. The formula to create the compound variable depends on the variable type; for example, two string variables can be merged into a new string by appending them; a new number variable can be created by the algebraic formula of two numerical variables.

Also, a compound variable can call NetBrain API to transfer a Parser variable to a new variable, for example, from an IP address to a hostname.



Name: Fix Interface ACL Property

Description:

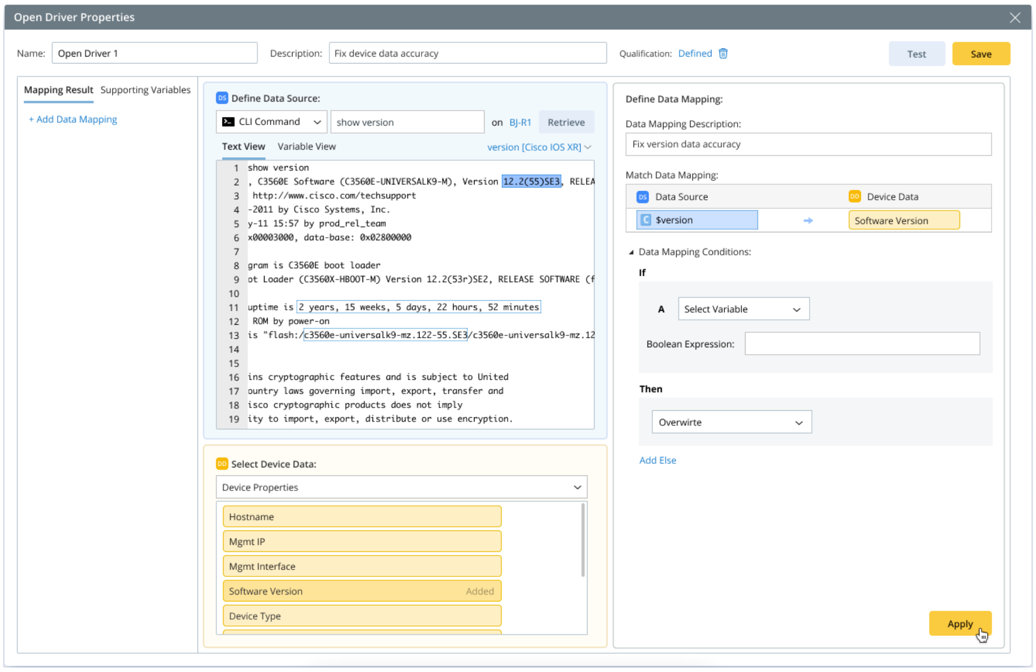
Qualification: Undefined

Define Data Source:

Please add a variable or select a variable to view details.

3.5.1.3 Map Data Source Variables to Target Device Data

Mapping a single value variable is common for most Device and interface GDRs. To define this data mapping, you select a data source variable and a target device data and define the condition. By default, there is no condition, and the system simply assigns the data source variable to the target device data (overwrite) even if the value is empty.


Open Driver Properties

Name: Open Driver 1 Description: Fix device data accuracy Qualification: Defined

Define Data Source: CLI Command show version on BJ-R1

version [Cisco IOS XR]

```
show version
, C3560E Software (C3560E-UNIVERSALK9-M), Version 12.2(55)SE3, RELEA
 http://www.cisco.com/techsupport
-2011 by Cisco Systems, Inc.
y-11 15:57 by prod_rel_team
x00003000, data-base: 0x02800000

gram is C3560E boot loader
ot Loader (C3560X-HBOOT-M) Version 12.2(53r)SE2, RELEASE SOFTWARE (f

uptime is 2 years, 15 weeks, 5 days, 22 hours, 52 minutes
 ROM by power-on
is "flash:/c3560e-universalk9-mz.122-55.SE3/c3560e-universalk9-mz.12


ins cryptographic features and is subject to United
ountry laws governing import, export, transfer and
isco cryptographic products does not imply
ity to import, export, distribute or use encryption.
```

Select Device Data: Device Properties

- Hostname
- Mgmt IP
- Mgmt Interface
- Software Version — Added
- Device Type

Define Data Mapping:

Data Mapping Description: Fix version data accuracy

Match Data Mapping:

Data Source		Device Data
$version	→	Software Version

Data Mapping Conditions:

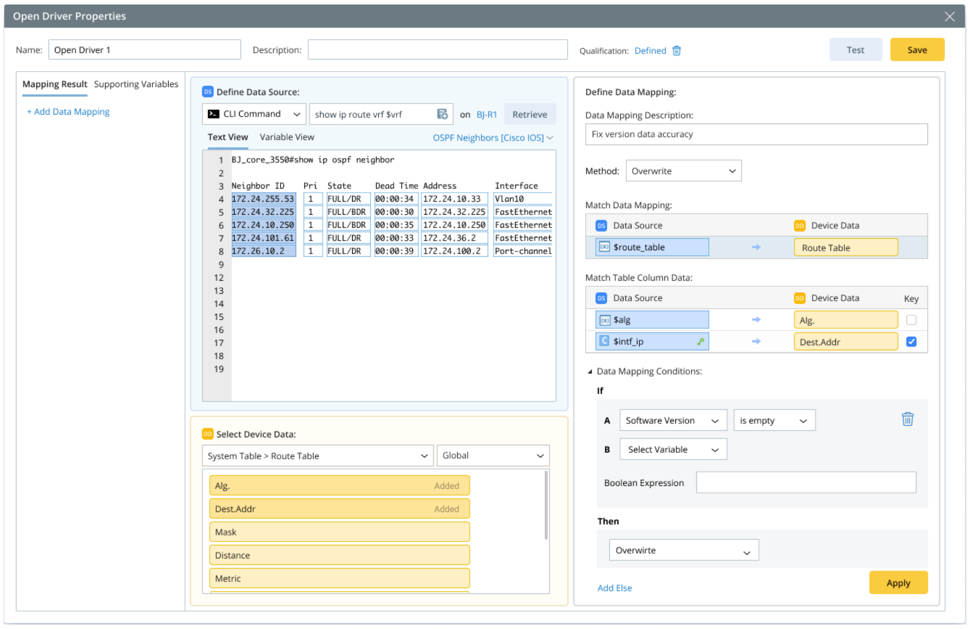
If

A: Select Variable

Boolean Expression:

Then: Overwirte

To map a table, the user needs to match the columns of the source table to the target table after selecting the source and target table. Also, he needs to define which column of the target table is the key if it is not defined. The keys will identify the corresponding rows of the source table and target table.


Open Driver Properties

Name: Open Driver 1 Description: Qualification: Defined

Define Data Source: CLI Command show ip route vrf $vrf on BJ-R1

OSPF Neighbors [Cisco IOS]

```
BJ_core_3550#show ip ospf neighbor

Neighbor ID     Pri  State      Dead Time Address        Interface
172.24.255.53   1    FULL/DR    00:00:34  172.24.10.33   Vlan10
172.24.32.225   1    FULL/BDR   00:00:30  172.24.32.225  FastEthernet
172.24.10.250   1    FULL/BDR   00:00:35  172.24.10.250  FastEthernet
172.24.101.61   1    FULL/DR    00:00:33  172.24.36.2    FastEthernet
172.26.10.2     1    FULL/DR    00:00:39  172.24.100.2   Port-channel
```

Select Device Data: System Table > Route Table | Global

- Alg. — Added
- Dest.Addr — Added
- Mask
- Distance
- Metric

Define Data Mapping:

Data Mapping Description: Fix version data accuracy

Method: Overwrite

Match Data Mapping:

Data Source		Device Data
$route_table	→	Route Table

Match Table Column Data:

Data Source		Device Data	Key
$alg	→	Alg.	☐
$intf_ip	→	Dest.Addr	☑

Data Mapping Conditions:

If

A: Software Version — is empty

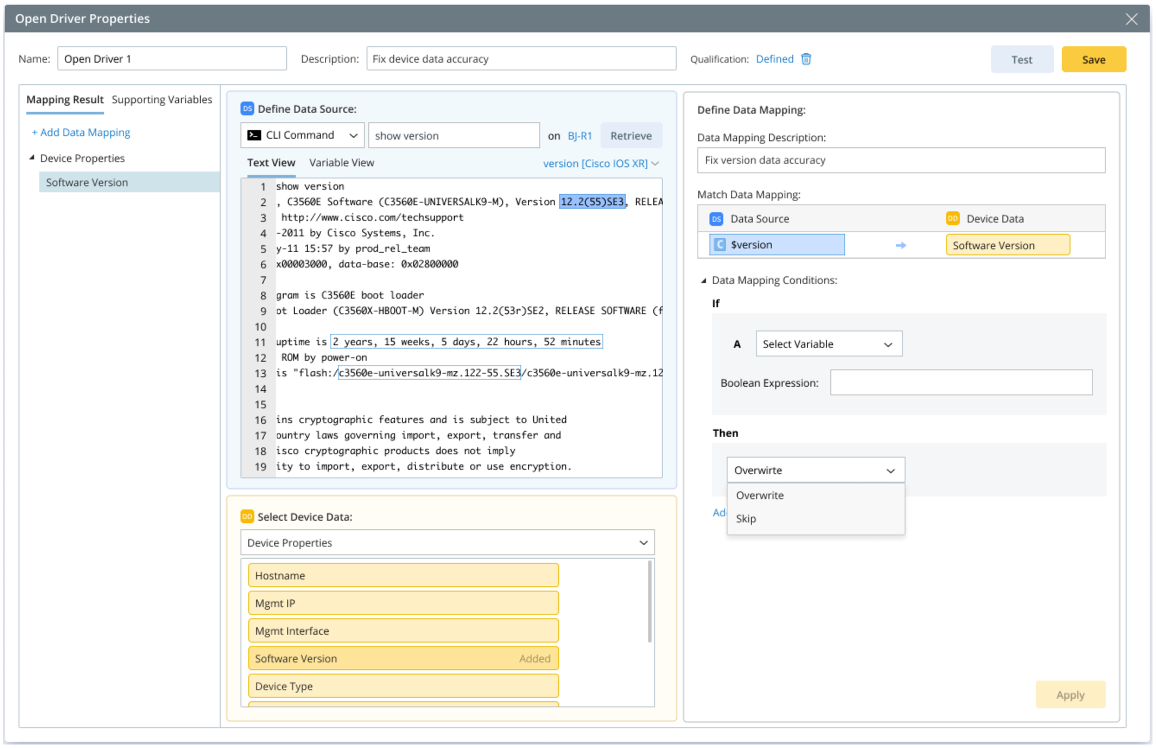
B: Select Variable

Boolean Expression

Then: Overwrite

There are three methods to map the tables: Overwrite (similar to Left Join of SQL), Stitch (similar to Right Join of SQL), and Merge (similar to Outer Join of SQL).

Users can define a mapping condition. A simple use case for the mapping condition is: if the source data variable is empty (the system cannot retrieve the data), do not overwrite the target device data. Also, the user can develop multiple ODs for one target device data (for example, one OD via SNMP and the other OD via CLI command to set a device location). And mapping condition can be used to define what condition an OD can overwrite the target device data.


Name: Open Driver 1
Description: Fix device data accuracy
Qualification: Defined

Define Data Source: CLI Command, show version on BJ-R1

```
show version
, C3560E Software (C3560E-UNIVERSALK9-M), Version 12.2(55)SE3, RELEA
 http://www.cisco.com/techsupport
-2011 by Cisco Systems, Inc.
y-11 15:57 by prod_rel_team
x00003000, data-base: 0x02800000

gram is C3560E boot loader
ot Loader (C3560X-HBOOT-M) Version 12.2(53r)SE2, RELEASE SOFTWARE (f

uptime is 2 years, 15 weeks, 5 days, 22 hours, 52 minutes
 ROM by power-on
is "flash:/c3560e-universalk9-mz.122-55.SE3/c3560e-universalk9-mz.12


ins cryptographic features and is subject to United
ountry laws governing import, export, transfer and
isco cryptographic products does not imply
ity to import, export, distribute or use encryption.
```

Select Device Data: Device Properties
- Hostname
- Mgmt IP
- Mgmt Interface
- Software Version — Added
- Device Type

Define Data Mapping:
Data Mapping Description: Fix version data accuracy

Match Data Mapping:

Data Source	Device Data
$version	Software Version

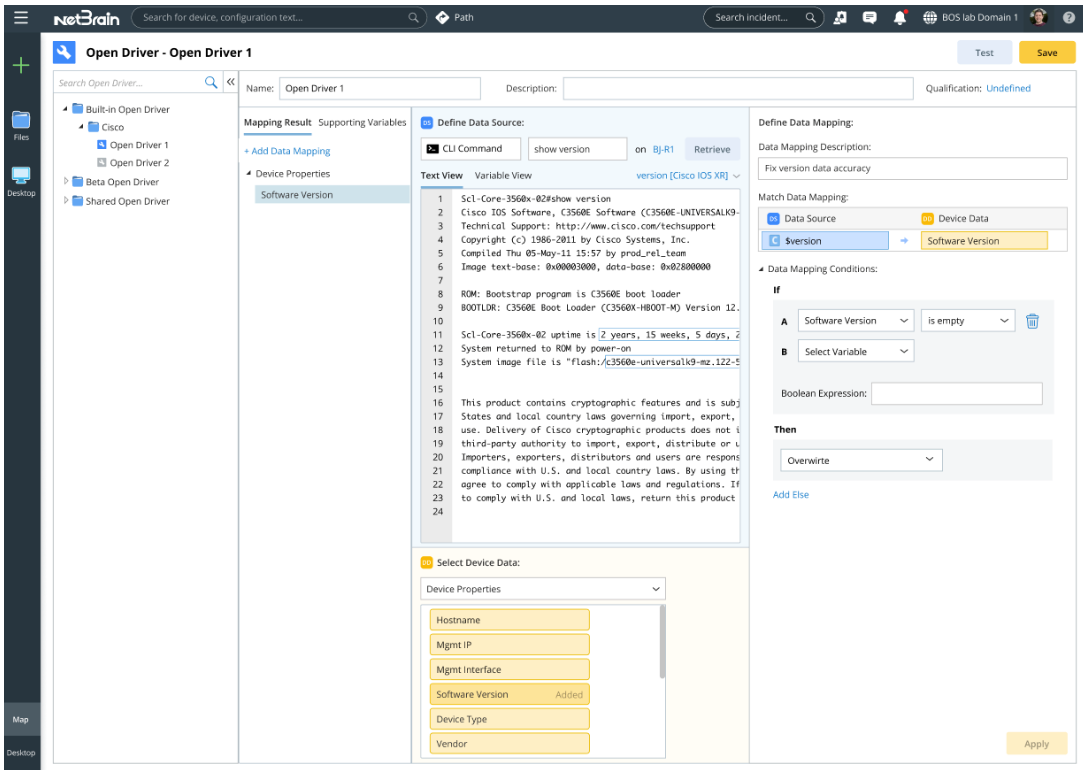
Data Mapping Conditions:
If
A: Select Variable
Boolean Expression:
Then: Overwirte
- Overwrite
- Skip

The actions for a single value variable can be **Overwrite** (assigning the source data to the target data) and **Skip** (do nothing).

The actions for a table column variable can be: **Overwrite; Overwrite if Source is Not Empty; Merge; Merge if Source is Not Empty; Skip.**

3.5.2 Test OD

OD Editor provides the Test button for the user to test run an OD.



Name: Open Driver 1

Description:

Qualification: Undefined

CLI Command: show version on BJ-R1 — Retrieve

version [Cisco IOS XR]

```
Scl-Core-3560x-02#show version
Cisco IOS Software, C3560E Software (C3560E-UNIVERSALK9-
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2011 by Cisco Systems, Inc.
Compiled Thu 05-May-11 15:57 by prod_rel_team
Image text-base: 0x00003000, data-base: 0x02800000

ROM: Bootstrap program is C3560E boot loader
BOOTLDR: C3560E Boot Loader (C3560X-HBOOT-M) Version 12.

Scl-Core-3560x-02 uptime is 2 years, 15 weeks, 5 days, 2
System returned to ROM by power-on
System image file is "flash:/c3560e-universalk9-mz.122-5


This product contains cryptographic features and is subj
States and local country laws governing import, export,
use. Delivery of Cisco cryptographic products does not i
third-party authority to import, export, distribute or u
Importers, exporters, distributors and users are respons
compliance with U.S. and local country laws. By using th
agree to comply with applicable laws and regulations. If
to comply with U.S. and local laws, return this product
```

Device Properties

Data Mapping Description: Fix version data accuracy

Data Source	Device Data
$version	Software Version

A: Software Version — is empty

B: Select Variable

Boolean Expression:

Then: Overwrite

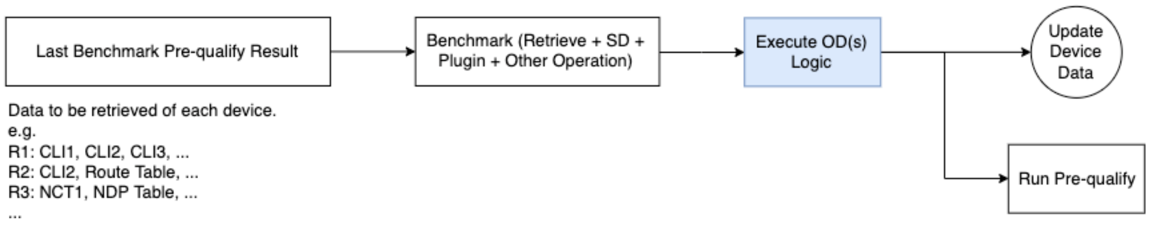
After the user selects a device to run the OD, the system will display the execution logs, showing which data is retrieved and which target device data is set.

3.5.3 Run OD

After the scheduled Platform Validation (PV) check, all data accuracy issues are displayed in the DAW pane, showing the error code, error message, and recommended action. Many of these issues will be solved through OD. For example, the interface column of an ARP table is empty. Without waiting for the NetBrain team to update SD, the user can add an OD to parse the ARP table correctly and modify the ARP table. After testing the OD on this device, the user can apply the OD to other qualified devices.

ODs will be automatically run after the corresponding SD in all functions, which will execute SD with one exception, Discovery. For example, Benchmark, Retrieve the device data, Path calculation, retrieve the system table and NCT table, Qapp, etc. If multiple ODs are enabled in a device, the system will execute ODs by order of the latest modified time (the most recently modified OD will be executed first).

To reduce the frequency to login to a device, the system executes ODs in two steps in the benchmark process:


Last Benchmark Pre-qualify Result
Benchmark (Retrieve + SD + Plugin + Other Operation)
Execute OD(s) Logic
Update Device Data
Run Pre-qualify
Data to be retrieved of each device.
e.g.
R1: CLI1, CLI2, CLI3, ...
R2: CLI2, Route Table, ...
R3: NCT1, NDP Table, ...
...

1. Retrieve all data on all qualified devices. The system will retrieve all data from the devices qualified for an OD according to the OD definition.

2. Parse variables and update target data.

3.5.4 Manage OD

The system provides an OD Management interface to view and manage all ODs under domain management. ODs has three root nodes according to their type:

- Built-in ODs: ODs created by the NetBrain team and will be updated via Knowledge Cloud (KC).

- Beta ODs: ODs created by the NetBrain team and will be updated via KC. V10.1 introduces the concept of beta SD and beta OD. All beta ODs are not enabled to run by default. The user can test run a beta OD and enable this OD if the results are satisfying. After a beta OD has been tested successfully with customers, the NetBrain team will move a beta OD to a built-in OD so that all customers can benefit from it.

- Customer-defined ODs: the ODs created and managed by the customer. The customer can create a new folder under this root node and create/edit/delete an OD.

The built-in and beta ODs operations are primarily reserved for the NetBrain Platform team. A unique username and password will be required to enable the operations. Or, the system administrator enables the debug mode to run these operations, including the import/export, create/edit/delete, etc.

4 New Foundation Modules

4.1 Google Cloud Visual Management

As more and more IT workloads are being moved to the public cloud like Google Cloud Platform (GCP), operating the public Cloud like GCP becomes challenging for IT specialists. Even if the automation and agility during the provisioning process have been greatly improved, it is not the same when it comes to the manageability of the public Cloud. The main challenges of managing a public cloud consist of the following aspects:

- **Lack of visibility**: agile provision of cloud resources makes visibility difficult using the traditional manual method.
- **A huge number of accounts and subscriptions**: to comply with the security requirements, you may have many accounts and subscriptions used by different teams. Managing all resources scattered in these accounts and subscriptions brings a heavy management burden to the team to troubleshoot cloud issues.
- **Multi-cloud and hybrid-cloud environments**: East-West Traffic supporting key applications often traverse physical data centers, SDN data centers, and public Clouds like Google Cloud. You may also have different public clouds such as AWS, Azure, and GCP to prevent vendor lock-in. As a result, many organizations bring multiple public clouds into their production use, and you need to understand different cloud providers' uniqueness.
- **Collaboration within different teams and customers**: an application that traverses through your network may involve multiple teams: network team, security team, cloud team, service team, and application team. When a problem occurs, you may need to involve all related teams to determine the root cause.

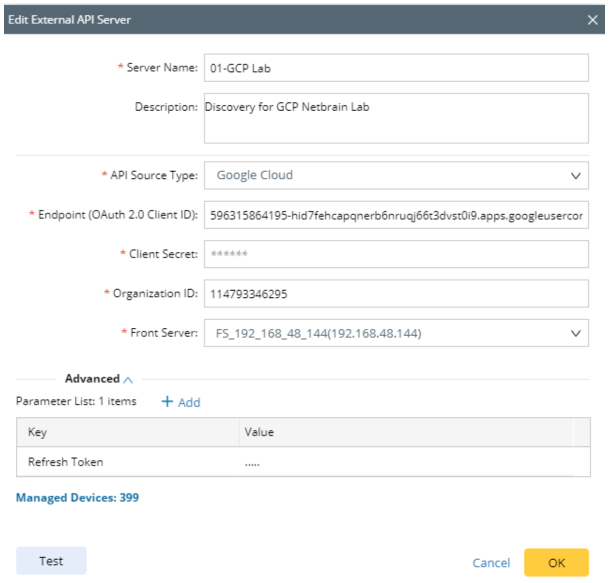
The need to have visibility into the public Cloud like GCP becomes critical. Therefore, in V10.1, we have built the support for GCP consisting of the following areas:

- **Auto Discovery:** NetBrain can discover Google Cloud resources and update the data periodically by leveraging the benchmark function.
- **Review network data and config with dynamic mapping**: As we do for traditional and SDN networks, we use API to access GCP and provide the data model. You can build the map based on the data model. The system can periodically retrieve the data from Google Cloud providers and update the data model accordingly.

- **Map application dependency across the end-to-end network**: with the ability to build the data model for public Cloud, SDN, SD-WAN, and the traditional network, NetBrain can provide you the path analysis function across the entire network. NetBrain can check the routing table, security group, network ACL for all the networking objects along the path and display the checking result details.
- **SPOG access through cloud-native and 3rd party cloud management tools**: NetBrain can display the cloud infrastructure data via Data View Templated (DVT) from GCP, Azure, and AWS API, display the cloud monitoring data from GCP Metric and Logs, AWS Cloud Watch, and Azure monitoring. Also, NetBrain can integrate with the cloud monitoring tools, such as Datadog, Splunk, and Dynatrace, to overlay the monitoring metrics/logging information on the NetBrain map.
- **Automate Troubleshooting with Runbook**: The support for Runbook Automation is also expanded to the support of the public Cloud like GCP. You can build different Runbooks according to different troubleshooting scenarios and leverage the Automation within NetBrain's entire automation reference workflow.

4.1.1 Discover GCP Resources

NetBrain uses the GCP endpoint to discover GCP resources via API.



* Server Name:	01-GCP Lab
Description:	Discovery for GCP Netbrain Lab
* API Source Type:	Google Cloud
* Endpoint (OAuth 2.0 Client ID):	596315864195-hid7fehcapqnerb6nruqj66t3dvst0i9.apps.googleusercor
* Client Secret:	******
* Organization ID:	114793346295
* Front Server:	FS_192_168_48_144(192.168.48.144)

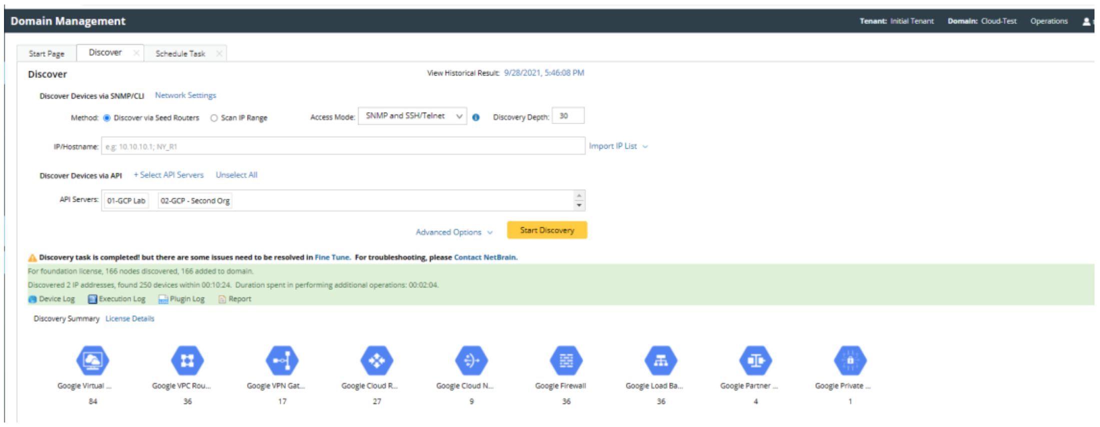
Advanced

Parameter List: 1 items + Add

Key	Value
Refresh Token	.....

Managed Devices: 399

Users can run the discovery task manually or schedule a discovery task to discover GCP resources.


Domain Management

Tenant: Initial Tenant   Domain: Cloud-Test   Operations

Start Page | Discover | Schedule Task

Discover

View Historical Result: 9/28/2021, 5:46:08 PM

Discover Devices via SNMP/CLI   Network Settings

Method: [✓ Discover via Seed Routers] Scan IP Range   Access Mode: SNMP and SSH/Telnet   Discovery Depth: 30

IP/Hostname: e.g: 10.10.10.1; NY_R1   Import IP List

Discover Devices via API   + Select API Servers   Unselect All

API Servers: 01-GCP Lab   02-GCP - Second Org

Advanced Options   Start Discovery

Discovery task is completed! but there are some issues need to be resolved in Fine Tune. For troubleshooting, please Contact NetBrain.

For foundation license, 166 nodes discovered, 166 added to domain.

Discovered 2 IP addresses, found 250 devices within 00:10:24. Duration spent in performing additional operations: 00:02:04.

Device Log   Execution Log   Plugin Log   Report

Discovery Summary   License Details

Google Virtual ...	Google VPC Rou...	Google VPN Gat...	Google Cloud R...	Google Cloud N...	Google Firewall	Google Load Ba...	Google Partner ...	Google Private ...
84	36	17	27	9	36	36	4	1

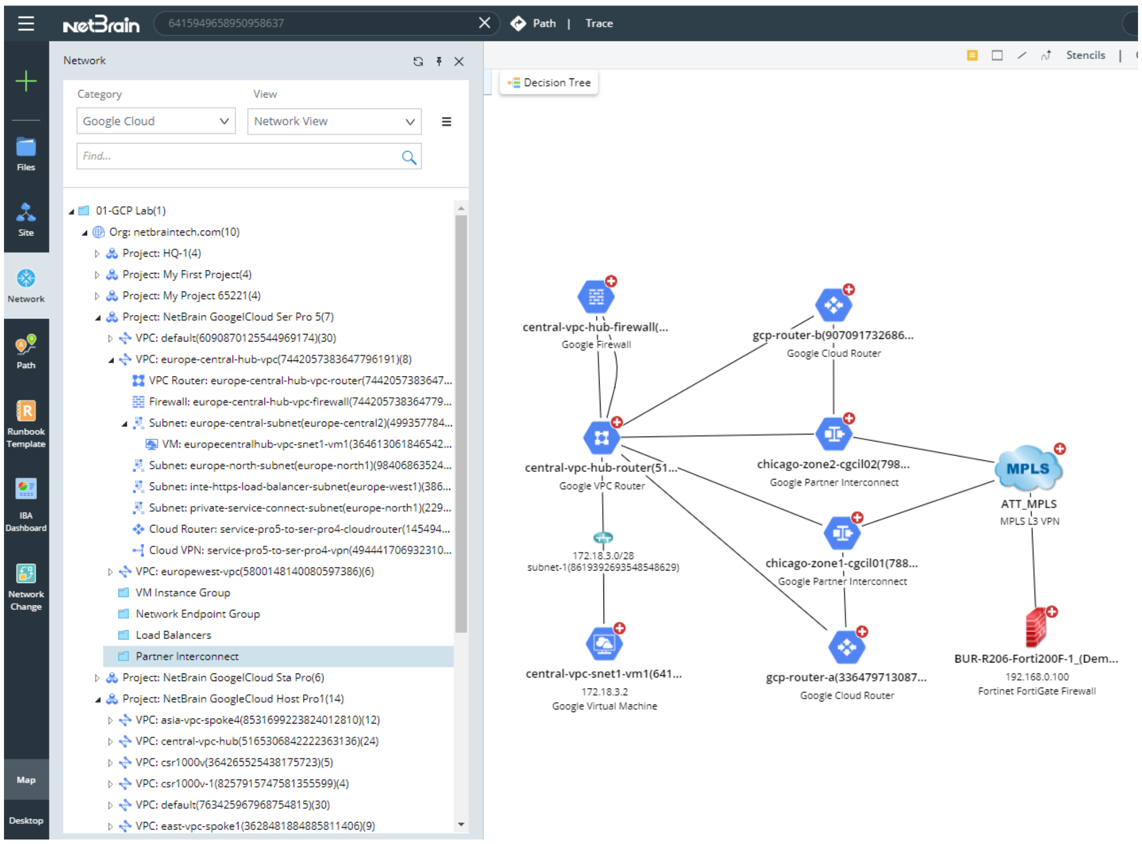
The discovered objects are displayed on GCP Network Tree, a hieratical view, Organization > Folder > Project > VPC Network. Through this tree, you have access to all accounts in a single view.

- **Level 1: Google Organization**
  - **Level 2: Google Folder**
    - **Level 2/1: Google Folder**
    - **Level 2/2: Google Folder**
      - **......**
        - **Level 2/2/.../n: Google Folder**
          - **Level 3: Google Project**
            - **Level 4: Google VPC Network**
              - **Level 5: Google VPC Router**
              - **Level 5: Google Subnet**
                - **Level 6: Google VM instance 1 (Folder)**
                - **Level 6: Google VM instance 2 (Folder)**
              - **Level 5: Google Cloud Router**
              - **Level 5: Google VPN Gateway**
              - **Level 5: Google Cloud NAT**
              - **Level 5: Google Firewall**
              - **Level 5: Google Network Virtual Appliance (NVA)**
                - **Level 6: Google VM instance**
            - **Level 4: Google VM Instance Group (Folder)**
            - **Level 4: Google Network Endpoint Group (Folder)**
            - **Level 4: Google Load Balancer (Folder)**
            - **Level 4: Google Partner Interconnect (Folder)**
            - **Level 4: Google Dedicated Interconnect (Folder)**

The network objects are listed based on their hierarchy, for example,

- The subnet is listed under the associated VPC Network.

- VM Instance is listed under the connected Subnet.
- Load Balancer is a logic object connected with different VPC networks in multiple regions under the project.
- Interconnect is not associated with a specific VPC Network, so it is under the project.


6415949658950958637
Path
Trace
Network
Category
View
Stencils
Decision Tree
Files
Site
Network
Path
Runbook Template
IBA Dashboard
Network Change
Map
Desktop
central-vpc-hub-firewall(...
Google Firewall
gcp-router-b(907091732686...
Google Cloud Router
central-vpc-hub-router(51...
Google VPC Router
chicago-zone2-cgcil02(798...
Google Partner Interconnect
MPLS
ATT_MPLS
MPLS L3 VPN
172.18.3.0/28
subnet-1(8619392693548548629)
chicago-zone1-cgcil01(788...
Google Partner Interconnect
central-vpc-snet1-vm1(641...
172.18.3.2
Google Virtual Machine
gcp-router-a(336479713087...
Google Cloud Router
BUR-R206-Forti200F-1_(Dem...
192.168.0.100
Fortinet FortiGate Firewall

Category: Google Cloud
View: Network View

- 01-GCP Lab(1)
  - Org: netbraintech.com(10)
    - Project: HQ-1(4)
    - Project: My First Project(4)
    - Project: My Project 65221(4)
    - Project: NetBrain GoogelCloud Ser Pro 5(7)
      - VPC: default(6090870125544969174)(30)
      - VPC: europe-central-hub-vpc(7442057383647796191)(8)
        - VPC Router: europe-central-hub-vpc-router(7442057383647...
        - Firewall: europe-central-hub-vpc-firewall(744205738364779...
        - Subnet: europe-central-subnet(europe-central2)(499357784...
          - VM: europecentralhub-vpc-snet1-vm1(364613061846542...
        - Subnet: europe-north-subnet(europe-north1)(98406863524...
        - Subnet: inte-https-load-balancer-subnet(europe-west1)(386...
        - Subnet: private-service-connect-subnet(europe-north1)(229...
        - Cloud Router: service-pro5-to-ser-pro4-cloudrouter(145494...
        - Cloud VPN: service-pro5-to-ser-pro4-vpn(49444170693231O...
      - VPC: europewest-vpc(5800148140080597386)(6)
      - VM Instance Group
      - Network Endpoint Group
      - Load Balancers
      - Partner Interconnect
    - Project: NetBrain GoogelCloud Sta Pro(6)
    - Project: NetBrain GoogleCloud Host Pro1(14)
      - VPC: asia-vpc-spoke4(8531699223824012810)(12)
      - VPC: central-vpc-hub(5165306842222363136)(24)
      - VPC: csr1000v(3642655254381757 23)(5)
      - VPC: csr1000v-1(8257915747581355599)(4)
      - VPC: default(7634259679687548 15)(30)
      - VPC: east-vpc-spoke1(3628481884885811406)(9)

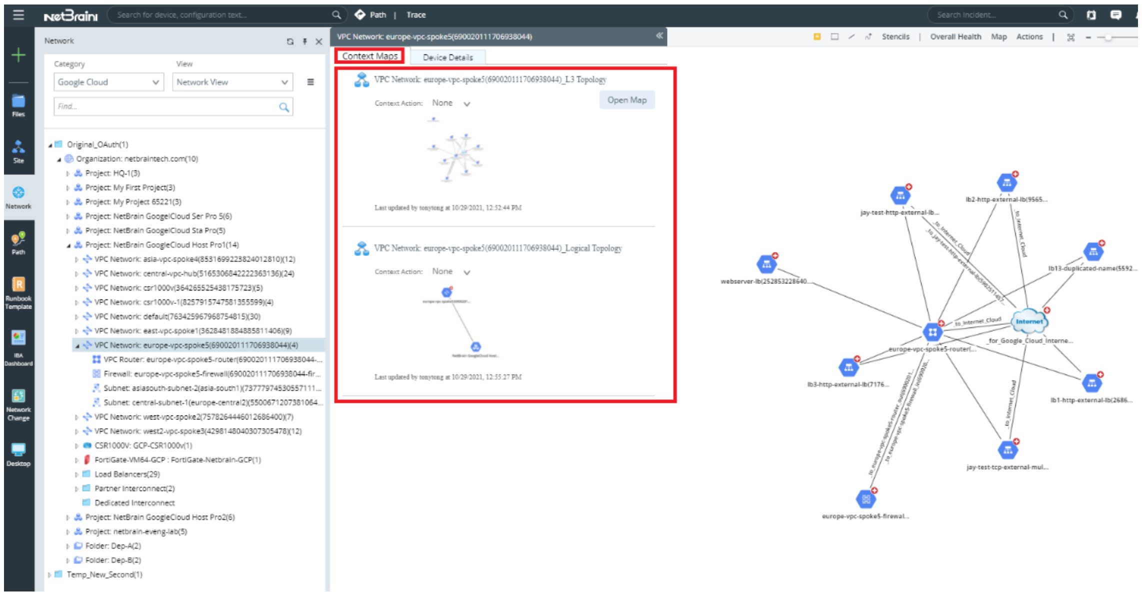
You can click an object from the network tree to view its detailed info, and the hyperlink will bring you to the GCP page for this object.

4.1.2 Map GCP Resources

There are different ways to map a GCP resource: open a context map of a resource from the Network Tree, search a resource via its IP or ID, and map it from the research results. You can extend neighbors of GCP resources like the on-premises network devices.

4.1.2.1 GCP Context Map Examples

A VPC Context Map demonstrates the Network reachability of a VPC Network:


Search for device, configuration text...
Path | Trace
Search Incident...
Stencils | Overall Health Map Actions
Files
Site
Network
Path
Runbook Template
IBA Dashboard
Network Change
Desktop
Open Map
jay-test-http-external-lb...
lb2-http-external-lb(9565...
webserver-lb(252853228640...
lb13-duplicated-name(5692...
Internet
europe-vpc-spoke5-router(...
lb3-http-external-lb(7176...
lb1-http-external-lb(2686...
jay-test-tcp-external-mul...
europe-vpc-spoke5-firewal...

Network

Category: Google Cloud
View: Network View

Find...

- Original_OAuth(1)
  - Organization: netbraintech.com(10)
    - Project: HQ-1(3)
    - Project: My First Project(3)
    - Project: My Project 65221(3)
    - Project: NetBrain GoogelCloud Ser Pro 5(6)
    - Project: NetBrain GoogelCloud Sta Pro(5)
    - Project: NetBrain GoogleCloud Host Pro1(14)
      - VPC Network: asia-vpc-spoke4(8531699223824012810)(12)
      - VPC Network: central-vpc-hub(5165306842222363136)(24)
      - VPC Network: csr1000v(3642655254381757 23)(5)
      - VPC Network: csr1000v-1(8257915747581355599)(4)
      - VPC Network: default(7634259679687548 15)(30)
      - VPC Network: east-vpc-spoke1(3628481884885811406)(9)
      - VPC Network: europe-vpc-spoke5(6900201117069380 44)(4)
        - VPC Router: europe-vpc-spoke5-router(690020111706938044-...
        - Firewall: europe-vpc-spoke5-firewall(690020111706938044-fir...
        - Subnet: asiasouth-subnet-2(asia-south1)(737779745305571 11...
        - Subnet: central-subnet-1(europe-central2)(550067120738106 4...
      - VPC Network: west-vpc-spoke2(7578264446012686400)(7)
      - VPC Network: west2-vpc-spoke3(4298148040307305478)(12)
      - CSR1000V: GCP-CSR1000v(1)
      - FortiGate-VM64-GCP : FortiGate-Netbrain-GCP(1)
      - Load Balancers(29)
      - Partner Interconnect(2)
      - Dedicated Interconnect
    - Project: NetBrain GoogleCloud Host Pro2(6)
    - Project: netbrain-eveng-lab(5)
    - Folder: Dep-A(2)
    - Folder: Dep-B(2)
- Temp_New_Second(1)

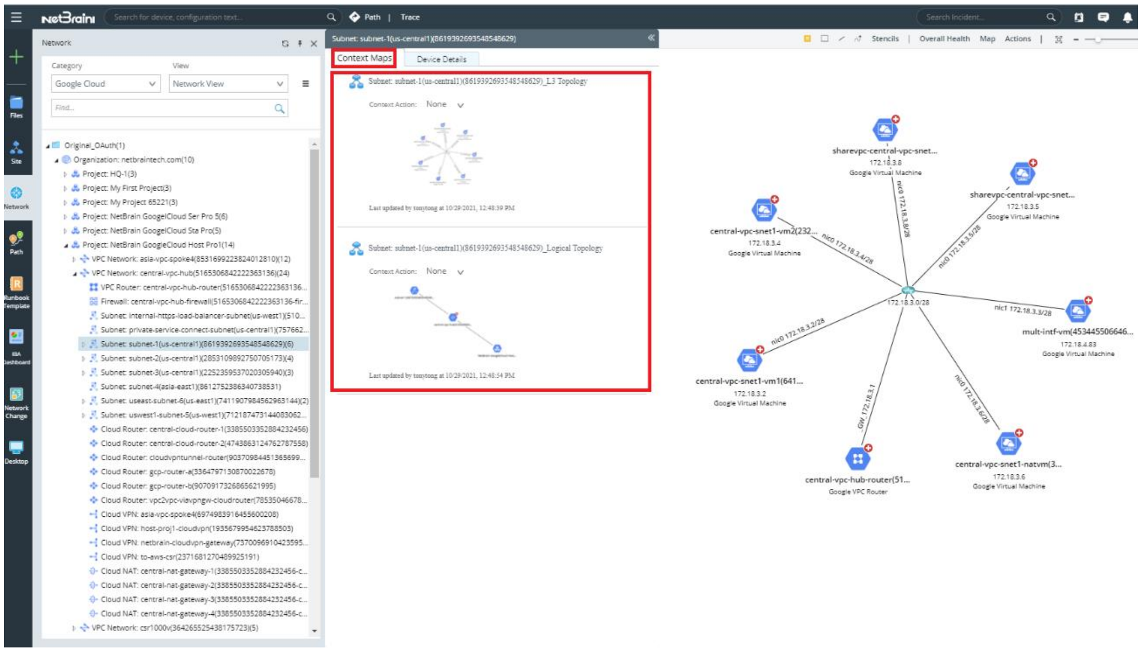
VPC Network: europe-vpc-spoke5(690020111706938044)

Context Maps | Device Details

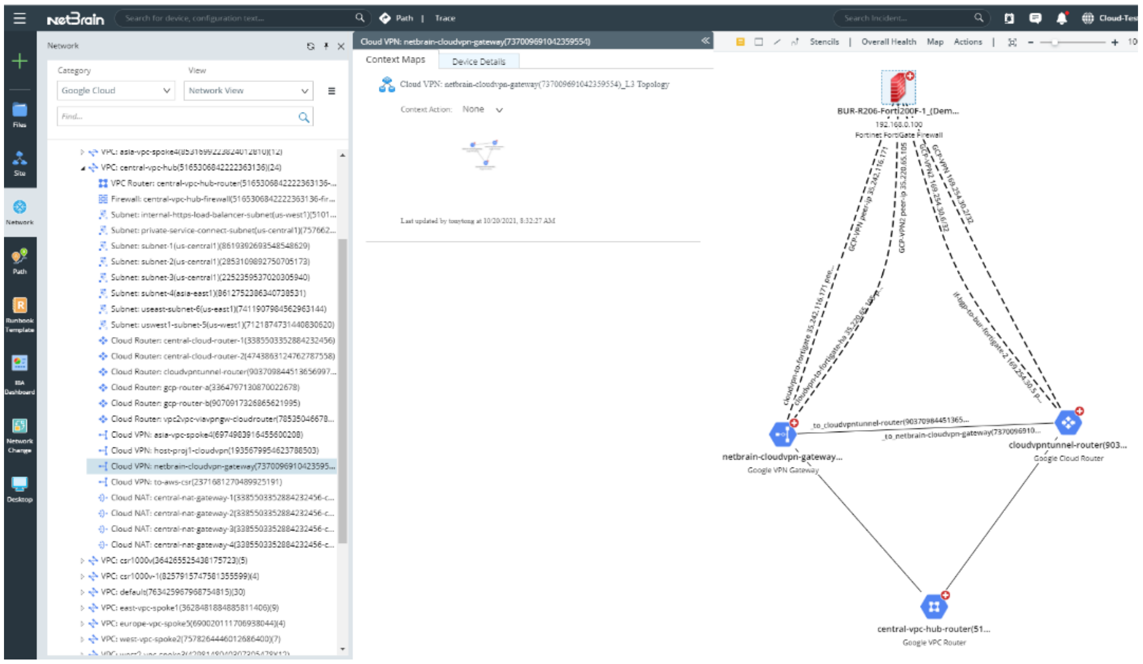
VPC Network: europe-vpc-spoke5(690020111706938044)_L3 Topology
Context Action: None
Last updated by tonytong at 10/29/2021, 12:52:44 PM

VPC Network: europe-vpc-spoke5(690020111706938044)_Logical Topology
Context Action: None
Last updated by tonytong at 10/29/2021, 12:55:27 PM

A subnet context map shows the virtual machine instances connected to the subnet within the VPC network.

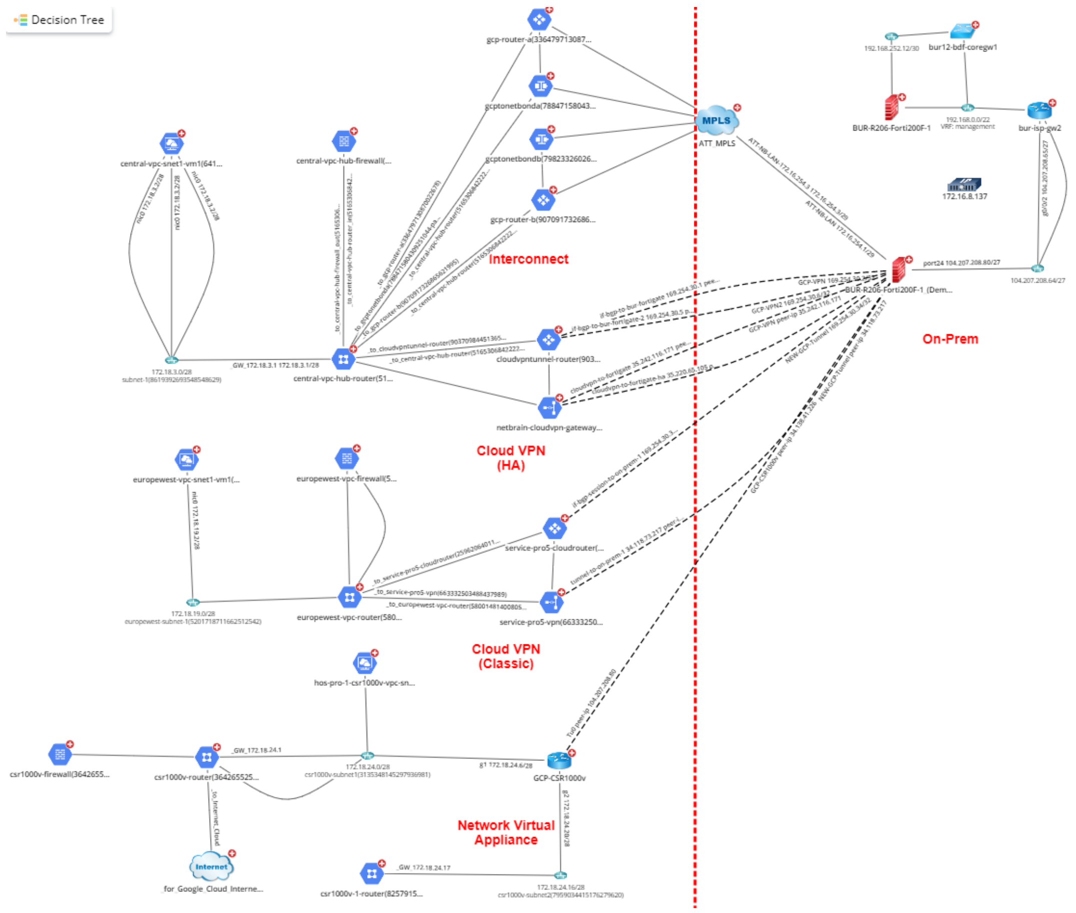


A VPN Gateway context map shows the relationship between Google Cloud and the on-premises network. The IPsec VPN connection between the VPN gateway and the on-promise edge device will be displayed. Also, the link for BGP Session between the Cloud Router and the on-premises edge device will be displayed.



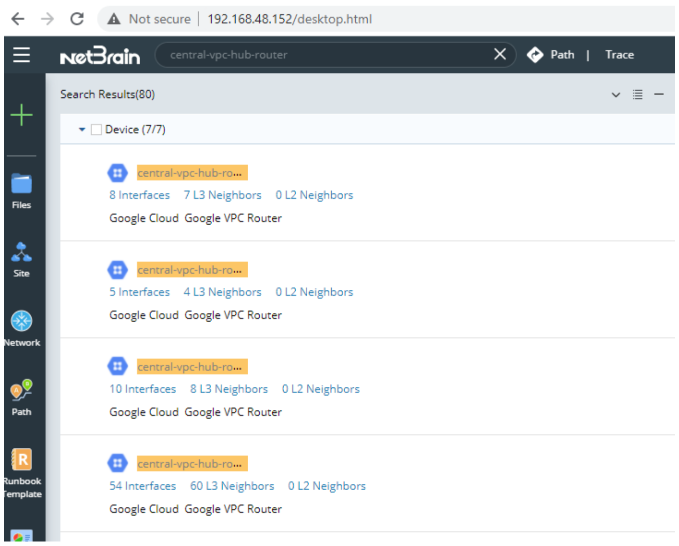
4.1.2.2 Hybrid Network Topology Map

Users can drag and drop a public cloud object or an on-premises network device on the map and extend its neighbors to map the connections between a public cloud and an on-premises network. Currently, we support the connection topology map and live path calculation between Google Cloud and On-Premises Network via **Partner/Dedicated Interconnect**, **HA/Classic VPN Gateway,** and **Network Virtual Appliance** hybrid solution.



4.1.2.3 Search and Map

You can search a public object with its name, IP address, or ID to identify where the resource is located and create maps based on the search results.



Search Results(80)

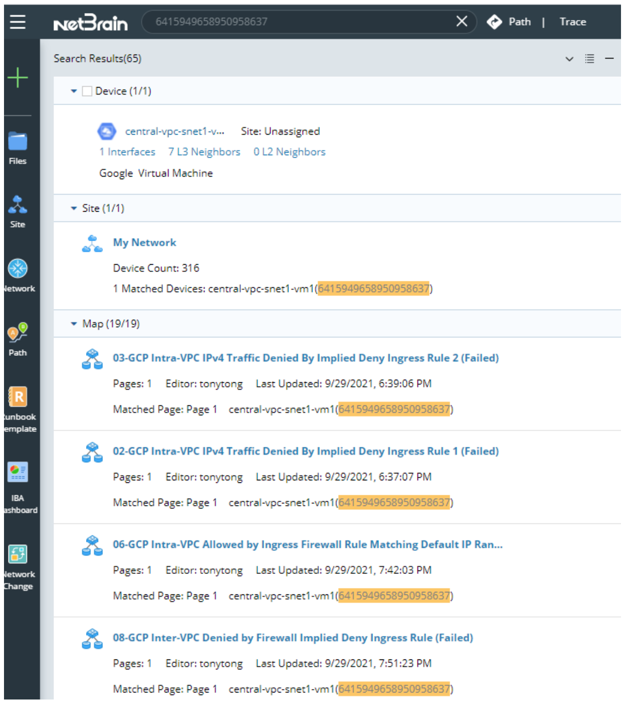
Device (7/7)

central-vpc-hub-ro...
8 Interfaces 7 L3 Neighbors 0 L2 Neighbors
Google Cloud Google VPC Router

central-vpc-hub-ro...
5 Interfaces 4 L3 Neighbors 0 L2 Neighbors
Google Cloud Google VPC Router

central-vpc-hub-ro...
10 Interfaces 8 L3 Neighbors 0 L2 Neighbors
Google Cloud Google VPC Router

central-vpc-hub-ro...
54 Interfaces 60 L3 Neighbors 0 L2 Neighbors
Google Cloud Google VPC Router



Search Results(65)

Device (1/1)

central-vpc-snet1-v... Site: Unassigned
1 Interfaces 7 L3 Neighbors 0 L2 Neighbors
Google Virtual Machine

Site (1/1)

My Network
Device Count: 316
1 Matched Devices: central-vpc-snet1-vm1(6415949658950958637)

Map (19/19)

03-GCP Intra-VPC IPv4 Traffic Denied By Implied Deny Ingress Rule 2 (Failed)
Pages: 1 Editor: tonytong Last Updated: 9/29/2021, 6:39:06 PM
Matched Page: Page 1 central-vpc-snet1-vm1(6415949658950958637)

02-GCP Intra-VPC IPv4 Traffic Denied By Implied Deny Ingress Rule 1 (Failed)
Pages: 1 Editor: tonytong Last Updated: 9/29/2021, 6:37:07 PM
Matched Page: Page 1 central-vpc-snet1-vm1(6415949658950958637)

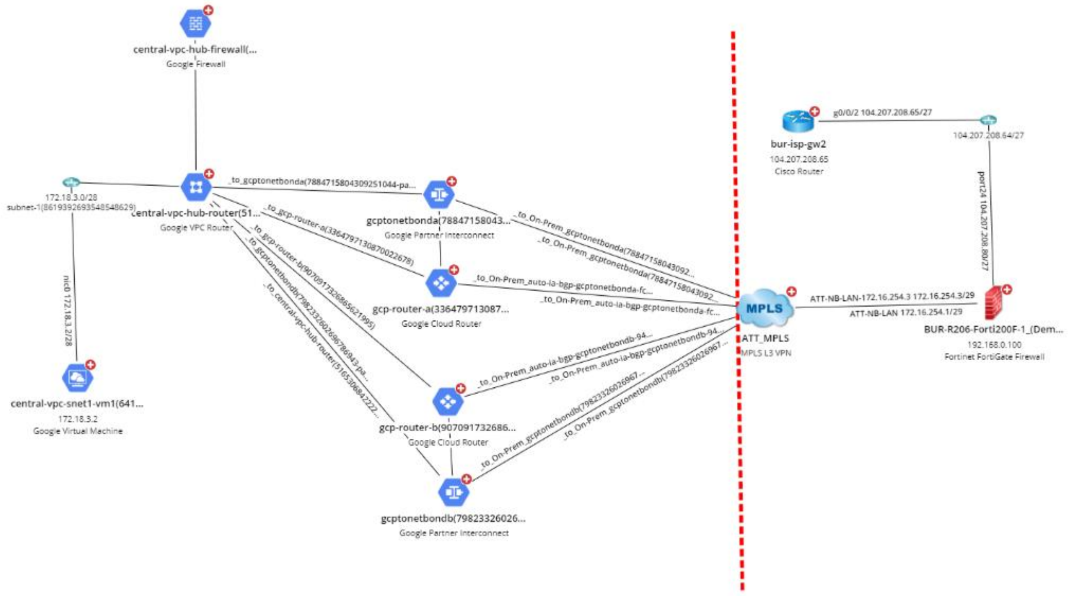
06-GCP Intra-VPC Allowed by Ingress Firewall Rule Matching Default IP Ran...
Pages: 1 Editor: tonytong Last Updated: 9/29/2021, 7:42:03 PM
Matched Page: Page 1 central-vpc-snet1-vm1(6415949658950958637)

08-GCP Inter-VPC Denied by Firewall Implied Deny Ingress Rule (Failed)
Pages: 1 Editor: tonytong Last Updated: 9/29/2021, 7:51:23 PM
Matched Page: Page 1 central-vpc-snet1-vm1(6415949658950958637)

4.1.3 Application Path for Hybrid and Multi-cloud

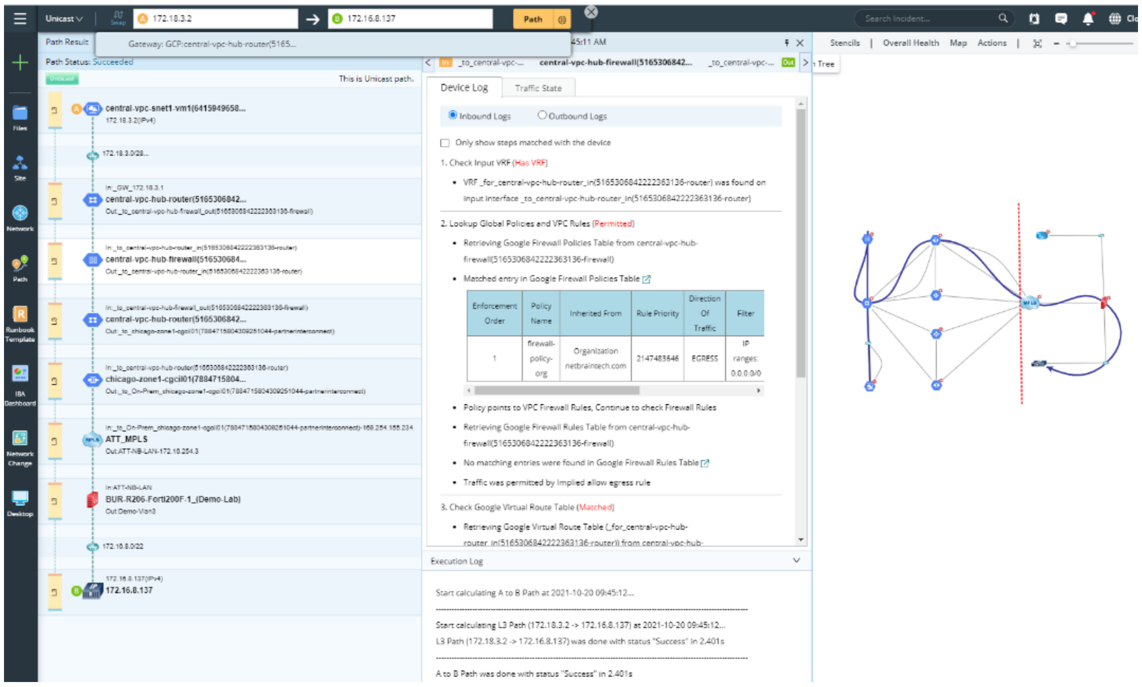
The path function has been extended to GCP. The system supports end-to-end path calculation in a hybrid and multi-cloud environment, and you can analyze the traffic flow between two endpoints.

A GCP VPC Network consists of IP range and subnets. It may also have cloud-native networking services such as VPN Gateway, Cloud NAT, Cloud Router, Cloud NAT, etc. NetBrain creates a VPC router for each VPC to simulate the routing and security check function. The GCP subnet is visualized in NetBrain's dynamic map via the LAN media concept so that you can view the different networking objects and understand how they are connected. VPC peering is also supported with the peering ID displayed on the map.



central-vpc-hub-firewall(...
Google Firewall
172.18.3.0/28
subnet-1(8619392693548548629)
central-vpc-hub-router(51...
Google VPC Router
_to_gcptonetbonda(7884715804309251044-pa...
gcptonetbonda(78847158043...
Google Partner Interconnect
gcp-router-a(336479713087...
Google Cloud Router
gcp-router-b(907091732686...
Google Cloud Router
gcptonetbondb(79823326026...
Google Partner Interconnect
nic0 172.18.3.2/28
central-vpc-snet1-vm1(641...
172.18.3.2
Google Virtual Machine
MPLS
ATT_MPLS
MPLS L3 VPN
bur-isp-gw2
104.207.208.65
Cisco Router
g0/0/2 104.207.208.65/27
104.207.208.64/27
port24 104.207.208.80/27
ATT-NB-LAN-172.16.254.3 172.16.254.3/29
ATT-NB-LAN 172.16.254.1/29
BUR-R206-Forti200F-1_(Dem...
192.168.0.100
Fortinet FortiGate Firewall

Path calculation will render the related routing and firewall rule/policy security check details.



Path Result

Gateway: GCP:central-vpc-hub-router(5165...

Path Status: Succeeded

Unicast

This is Unicast path.

central-vpc-snet1-vm1(6415949658...
172.18.3.2(IPv4)

172.18.3.0/28...

In:_GW_172.18.3.1
central-vpc-hub-router(5165306842...
Out:_to_central-vpc-hub-firewall_out(5165306842222363136-firewall)

In:_to_central-vpc-hub-router_in(5165306842222363136-router)
central-vpc-hub-firewall(516530684...
Out:_to_central-vpc-hub-router_in(5165306842222363136-router)

In:_to_central-vpc-hub-firewall_out(5165306842222363136-firewall)
central-vpc-hub-router(5165306842...
Out:_to_chicago-zone1-cgcil01(7884715804309251044-partnerinterconnect)

In:_to_central-vpc-hub-router(5165306842222363136-router)
chicago-zone1-cgcil01(7884715804...
Out:_to_On-Prem_chicago-zone1-cgcil01(7884715804309251044-partnerinterconnect)

In:_to_On-Prem_chicago-zone1-cgcil01(7884715804309251044-partnerinterconnect)-169.254.155.234
ATT_MPLS
Out:ATT-NB-LAN-172.16.254.3

In:ATT-NB-LAN
BUR-R206-Forti200F-1_(Demo-Lab)
Out:Demo-Vlan3

172.16.8.0/22

172.16.8.137(IPv4)
172.16.8.137

45:11 AM

In _to_central-vpc-... central-vpc-hub-firewall(5165306842... _to_central-vpc-... Out

Device Log | Traffic State

(●) Inbound Logs ( ) Outbound Logs

☐ Only show steps matched with the device

1. Check Input VRF (Has VRF)

- VRF _for_central-vpc-hub-router_in(5165306842222363136-router) was found on input interface _to_central-vpc-hub-router_in(5165306842222363136-router)

2. Lookup Global Policies and VPC Rules (Permitted)

- Retrieving Google Firewall Policies Table from central-vpc-hub-firewall(5165306842222363136-firewall)
- Matched entry in Google Firewall Policies Table

Enforcement Order	Policy Name	Inherited From	Rule Priority	Direction Of Traffic	Filter
1	firewall-policy-org	Organization netbraintech.com	2147483646	EGRESS	IP ranges: 0.0.0.0/0

- Policy points to VPC Firewall Rules, Continue to check Firewall Rules
- Retrieving Google Firewall Rules Table from central-vpc-hub-firewall(5165306842222363136-firewall)
- No matching entries were found in Google Firewall Rules Table
- Traffic was permitted by Implied allow egress rule

3. Check Google Virtual Route Table (Matched)

- Retrieving Google Virtual Route Table (_for_central-vpc-hub-router_in(5165306842222363136-router)) from central-vpc-hub-

Execution Log

```
Start calculating A to B Path at 2021-10-20 09:45:12...
------------------------------------------------------------
Start calculating L3 Path (172.18.3.2 -> 172.16.8.137) at 2021-10-20 09:45:12...
L3 Path (172.18.3.2 -> 172.16.8.137) was done with status "Success" in 2.401s
------------------------------------------------------------
A to B Path was done with status "Success" in 2.401s
```

4.1.3.1 Traffic across GCP and On-Premises Network

There are different ways to connect an on-premises network to a GCP VPC Network:

- **VPN Gateway Connect**

VPN Gateway securely connects your *peer network* to your Virtual Private Cloud (VPC) network through an IPsec VPN connection. Traffic traveling between the two networks is encrypted by one VPN gateway and then decrypted by the other. NetBrain supports visualizing the topology and path of VPN Gateway as well as the following data tables:

- ✓ Google Virtual Route Table
- ✓ Google Cloud VPN Tunnels Table

- **Partner/Dedicated Interconnect**

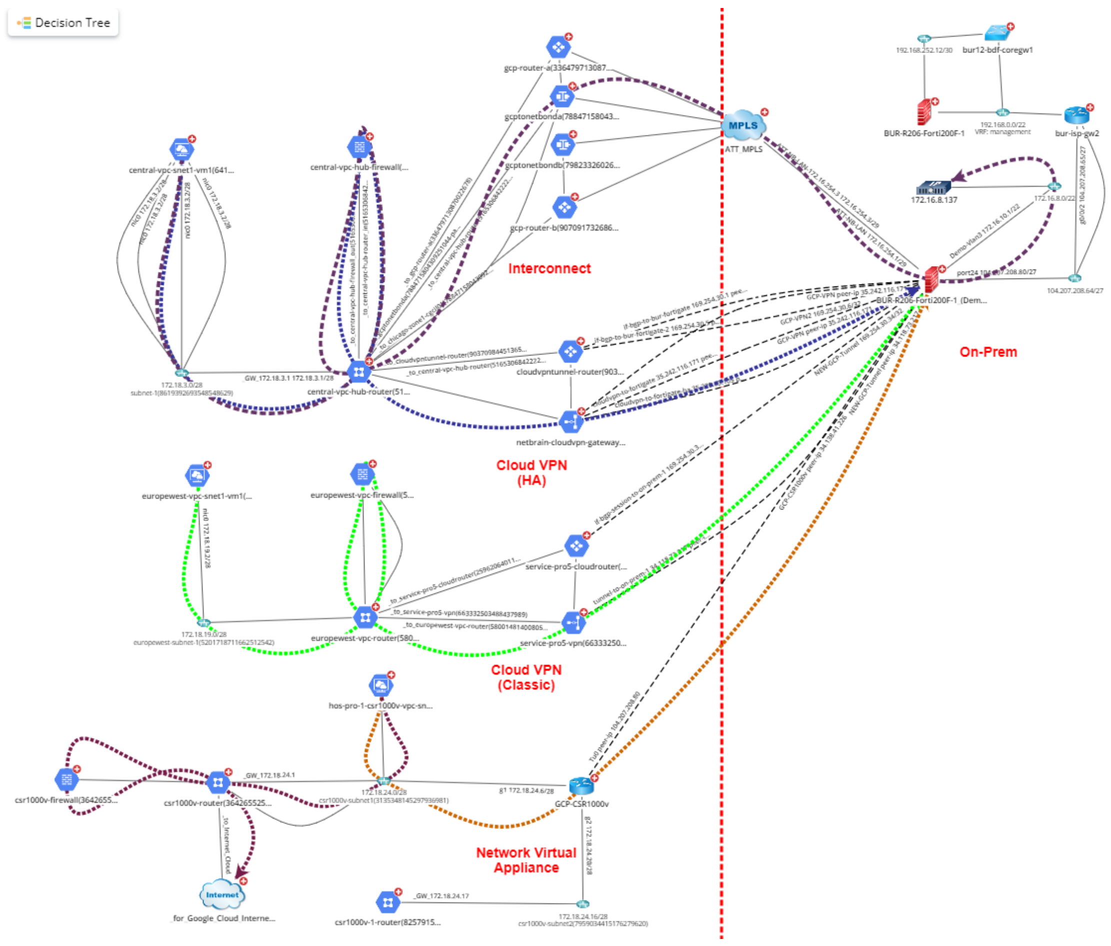
Cloud Interconnect extends your on-premises network to Google's network through a highly available, low latency connection. You can use Dedicated Interconnect to connect directly to GCP or use Partner Interconnect to connect to GCP through a supported service provider. NetBrain supports visualizing the topology and path of Partner/Dedicated Interconnect, as well as the following data tables:

- ✓ BGP Advertised Routes Table
- ✓ Google Partner Interconnect Physical Connections Table
- ✓ Google Partner Interconnect VLAN Attachment Table
- ✓ Google Virtual Route Table

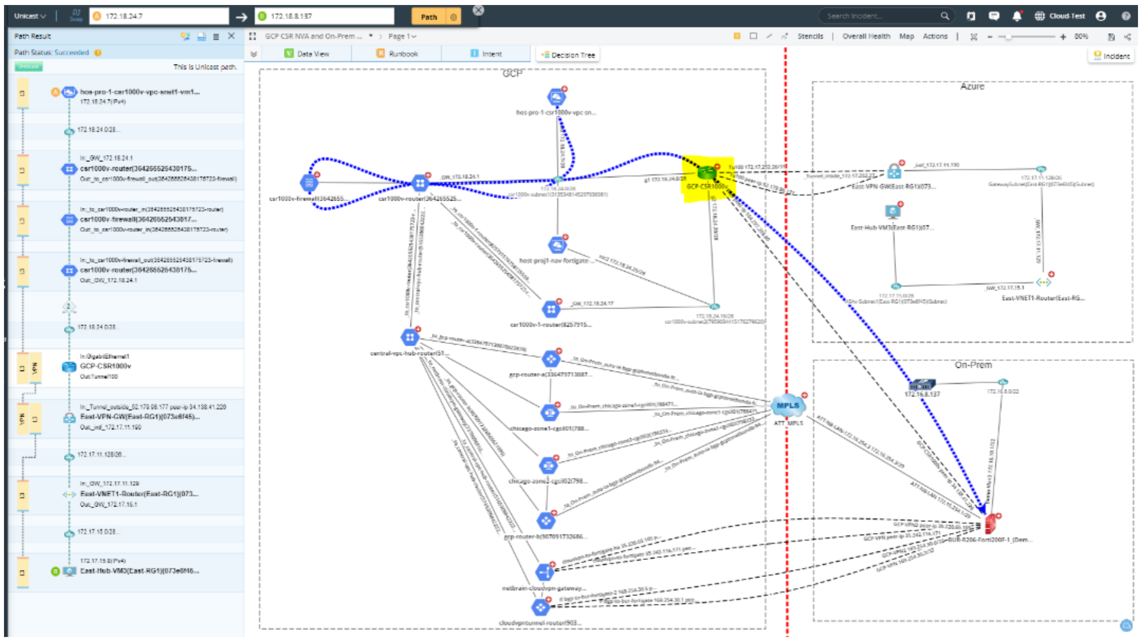
- **Network Virtual Appliance (NVA)**

NVA can be loaded with any vendor's virtual machine (VM) images to support networking, security, and other functions. NetBrain supports visualizing the topology and path of the VPN Tunnel connection between GCP NVA and On-Premises edge devices.

The following diagram demonstrates the path between GCP and the on-premises network, connected by the Interconnect, VPN, and VNA.

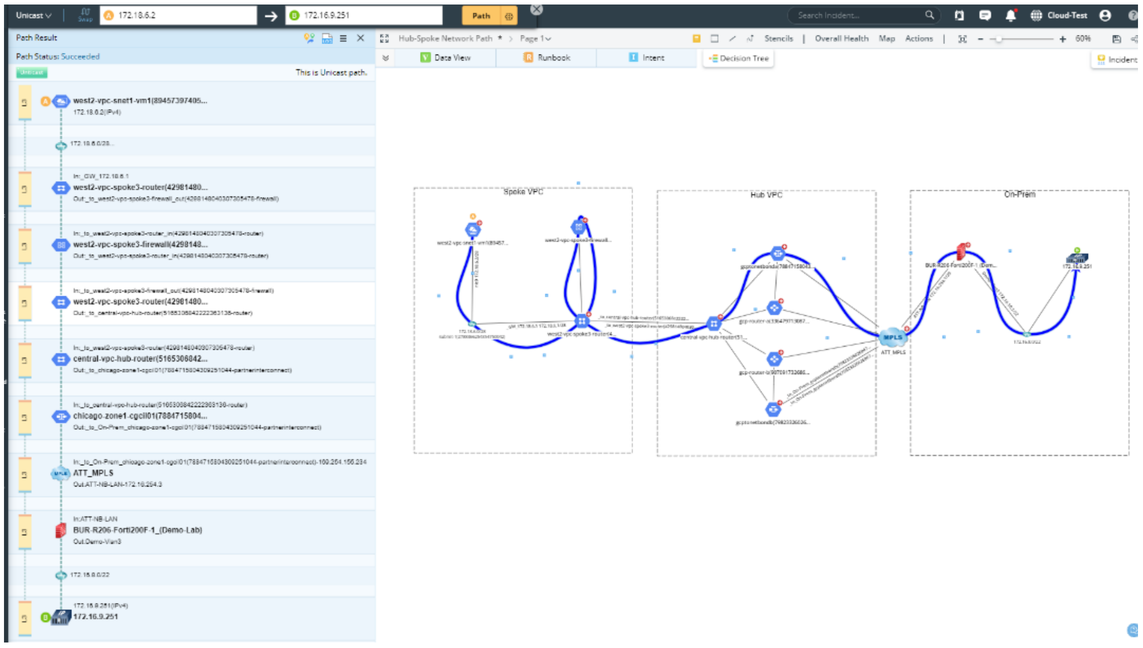


The following diagram shows the traffic through an example of VNA, a Cisco CSR 1000v Cloud Services Router, which provides a cloud-based virtual router deployed on a virtual machine (VM) instance on x86 server hardware.



4.1.3.2 Hub-Spoke Network

The Hub VPC Network in GCP acts as a central point of connectivity to your on-premises network. The spokes VPC Network has peer with the Hub. Shared services are deployed in the Hub, while individual workloads are deployed as spokes. The following path shows that the Hub provides a shared Interconnect resource for all Spokes VPC networks to visit the on-premises devices.



Path Result
Path Status: Succeeded
Unicast
This is Unicast path.

A west2-vpc-snet1-vm1(89457397405...
172.18.6.2(IPv4)

172.18.6.0/28...

In:_GW_172.18.6.1
west2-vpc-spoke3-router(42981480...
Out:_to_west2-vpc-spoke3-firewall_out(4298148040307305478-firewall)

In:_to_west2-vpc-spoke3-router_in(4298148040307305478-router)
west2-vpc-spoke3-firewall(4298148...
Out:_to_west2-vpc-spoke3-router_in(4298148040307305478-router)

In:_to_west2-vpc-spoke3-firewall_out(4298148040307305478-firewall)
west2-vpc-spoke3-router(42981480...
Out:_to_central-vpc-hub-router(5165306842222363138-router)

In:_to_west2-vpc-spoke3-router(4298148040307305478-router)
central-vpc-hub-router(5165306842...
Out:_to_chicago-zone1-cgcil01(7884715804309251044-partnerinterconnect)

In:_to_central-vpc-hub-router(5165306842222363138-router)
chicago-zone1-cgcil01(7884715804...
Out:_to_On-Prem_chicago-zone1-cgcil01(7884715804309251044-partnerinterconnect)

In:_to_On-Prem_chicago-zone1-cgcil01(7884715804309251044-partnerinterconnect)-169.254.156.234
ATT_MPLS
Out:ATT-NB-LAN-172.16.254.3

In:ATT-NB-LAN
BUR-R206-Forti200F-1_(Demo-Lab)
Out:Demo-Vlan3

172.16.8.0/22

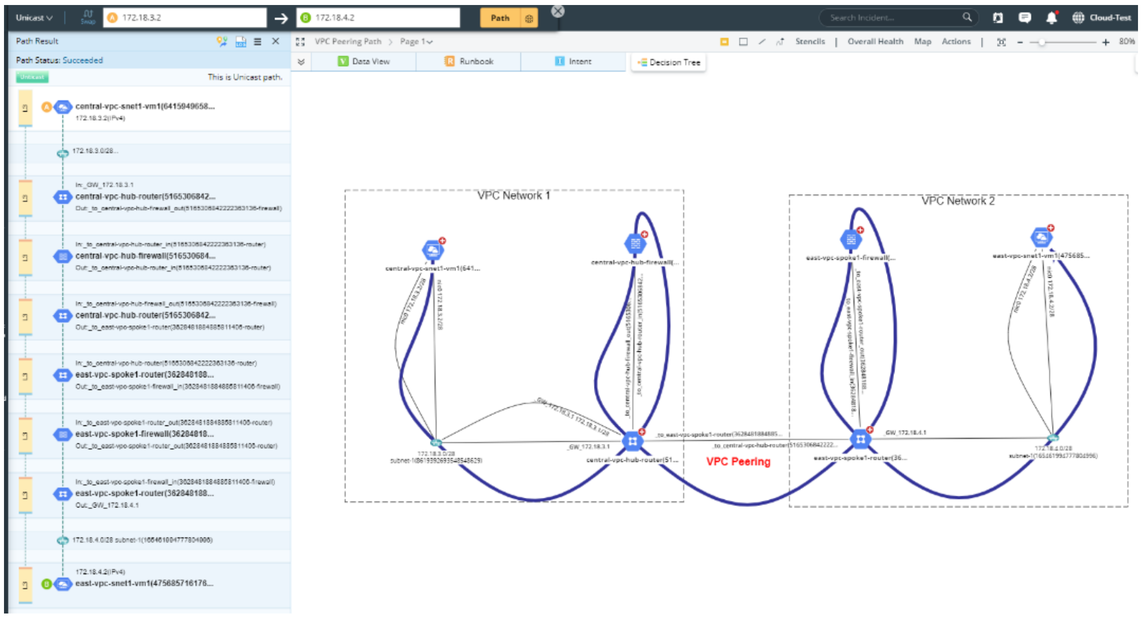
172.16.9.251(IPv4)
B 172.16.9.251

4.1.3.3 VPC Network Connect

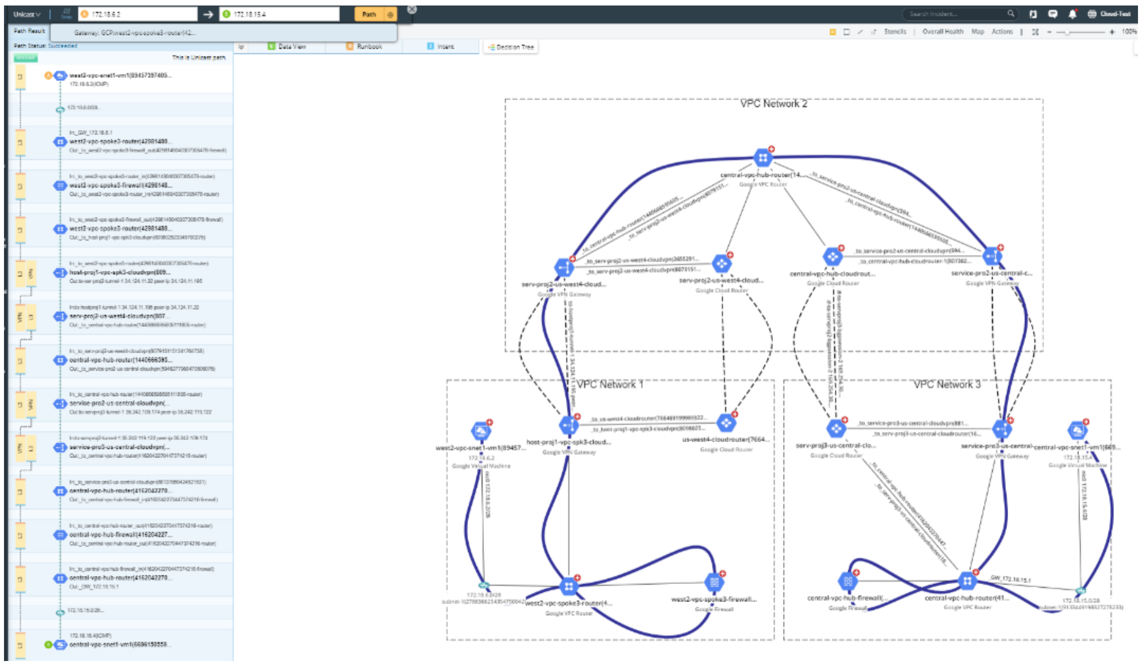
You can connect VPC networks with VPC peering or a VPN connection. NetBrain supports visualizing the topology and path of inter and intra VPC networks, as well as the following data tables:

- ✓ Google Virtual Route Table
- ✓ Google VPC Endpoint Group Table
- ✓ Google VPC Instance Group Table
- ✓ Google VPC Instance Group Members Table
- ✓ Google VPC Peering Table
- ✓ Google VPC Routes Table
- ✓ Google VPC Subnets Table

The following path demonstrates VPC network peering, which connects VPC networks so that workloads in different VPC networks can communicate internally. Thus, the traffic stays within the GCP and does not traverse the public internet.



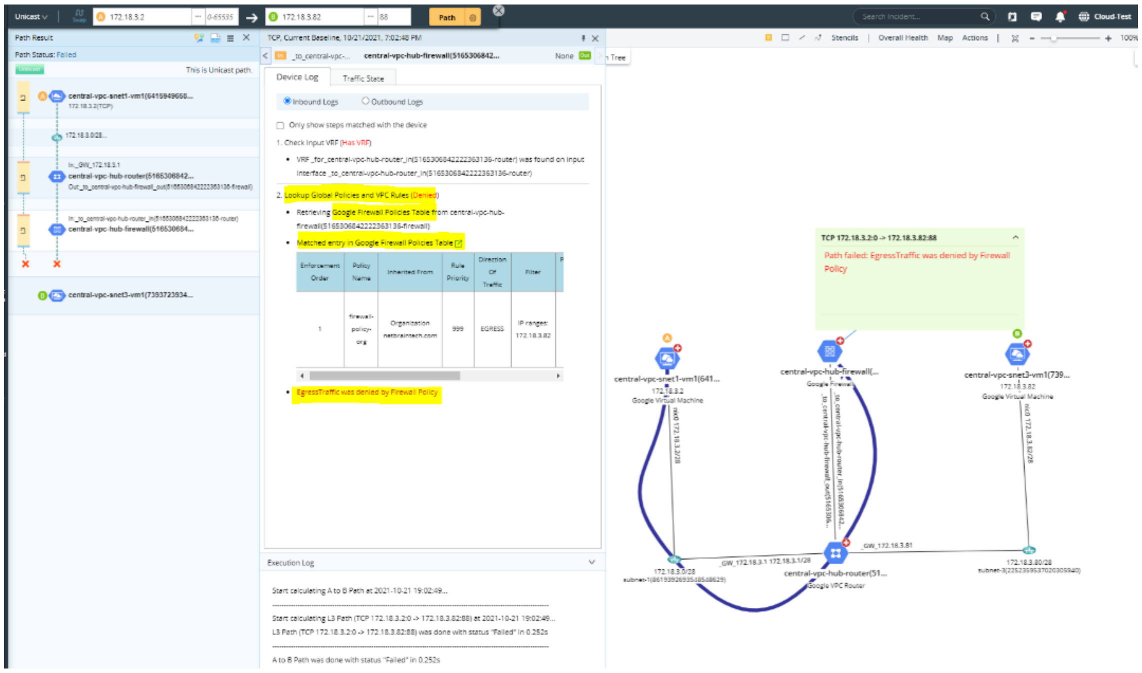
The following diagram demonstrates the path through an IPsec VPN connection. Traffic traveling between the two networks is encrypted by one VPN gateway and then decrypted by the other, protecting your data as it travels over the internet.



4.1.3.4 GCP Firewall Policy/Rule Check

NetBrain checks the GCP Firewall Policy and Rule while discovering the path through GCP objects. You can have a setting to bypass the Security Policy and Rule check. The following data tables are supported:

- ✓ Google Firewall Policies Table
- ✓ Google Firewall Rules Table
- ✓ Google Virtual Route Table



Unicast
172.18.3.2
0-65535
172.18.3.82
88
Path
Path Result
Path Status: Failed
This is Unicast path.
central-vpc-snet1-vm1(6415949658...
central-vpc-hub-router(5165306842...
central-vpc-hub-firewall(5165306684...
central-vpc-snet3-vm1(7393723934...
Stencils
Overall Health
Map
Actions
central-vpc-snet1-vm1(641...
172.18.3.2
Google Virtual Machine
central-vpc-hub-firewall(...
Google Firewall
central-vpc-snet3-vm1(739...
172.18.3.82
Google Virtual Machine
central-vpc-hub-router(51...
Google VPC Router

TCP, Current Baseline, 10/21/2021, 7:02:48 PM

_to_central-vpc-... central-vpc-hub-firewall(5165306842... None

Device Log | Traffic State

Inbound Logs | Outbound Logs

Only show steps matched with the device

1. Check Input VRF (Has VRF)
   - VRF _for_central-vpc-hub-router_in(5165306842222363136-router) was found on input interface _to_central-vpc-hub-router_in(5165306842222363136-router)

2. Lookup Global Policies and VPC Rules (Denied)
   - Retrieving Google Firewall Policies Table from central-vpc-hub-firewall(5165306842222363136-firewall)
   - Matched entry in Google Firewall Policies Table

Enforcement Order	Policy Name	Inherited From	Rule Priority	Direction Of Traffic	Filter
1	firewall-policy-org	Organization netbraintech.com	999	EGRESS	IP ranges: 172.18.3.82

   - EgressTraffic was denied by Firewall Policy

Execution Log

```
Start calculating A to B Path at 2021-10-21 19:02:49...
------------------------------------------------------------
Start calculating L3 Path (TCP 172.18.3.2:0 -> 172.18.3.82:88) at 2021-10-21 19:02:49...
L3 Path (TCP 172.18.3.2:0 -> 172.18.3.82:88) was done with status "Failed" in 0.252s
------------------------------------------------------------
A to B Path was done with status "Failed" in 0.252s
```

TCP 172.18.3.2:0 -> 172.18.3.82:88

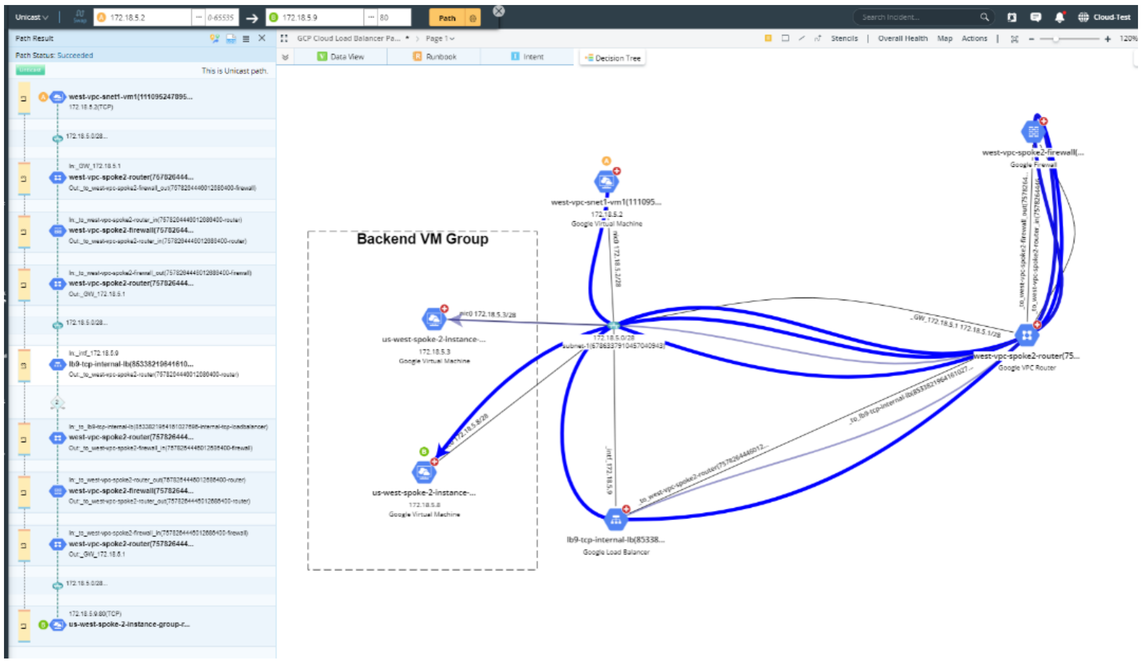
Path failed: EgressTraffic was denied by Firewall Policy

4.1.3.5 GCP Load Balancer

GCP Cloud Load Balancing is a fully distributed, software-defined managed service. NetBrain supports visualizing the topology and path of both External and Internal Load Balancer, as well as the following data tables:

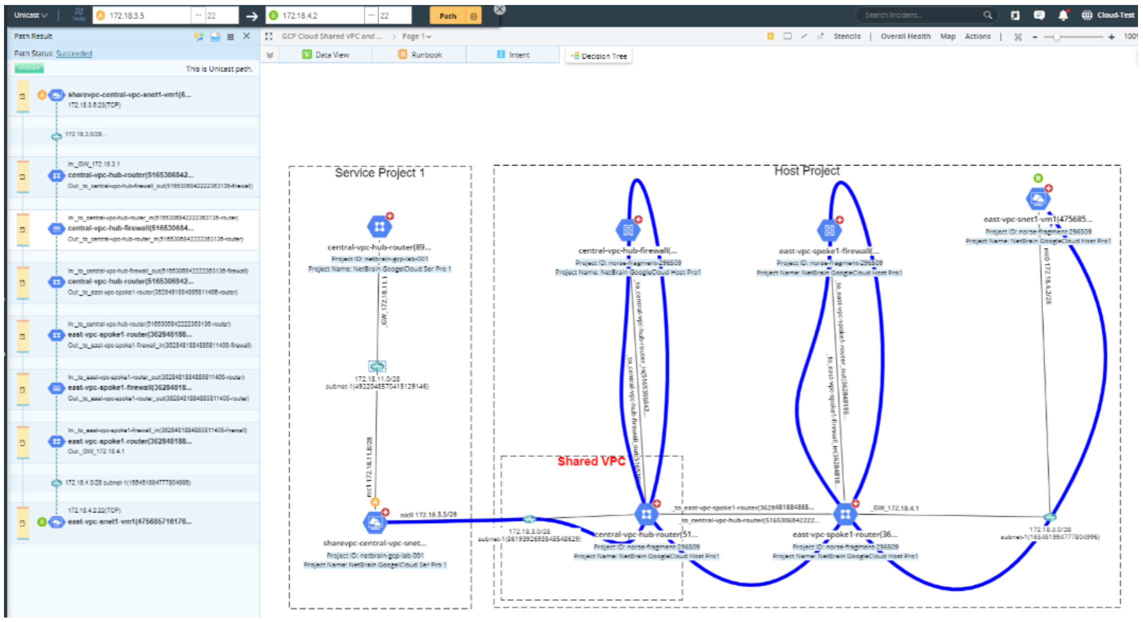
- ✓ Google Load Balancer Backend Table
- ✓ Google Load Balancer Forwarding Rules Table

- ✓ Google Load Balancer Host and Path Rules Table
- ✓ Google Virtual Route Table


Unicast
172.18.5.2
172.18.5.9
Path
Path Result
Path Status: Succeeded
This is Unicast path.
west-vpc-snet1-vm1(111095247895...
west-vpc-spoke2-router(757826444...
west-vpc-spoke2-firewall(7578264...
lb9-tcp-internal-lb(85338219641610...
us-west-spoke-2-instance-group-r...
GCP Cloud Load Balancer Pa...
Data View
Runbook
Intent
Decision Tree
Stencils
Overall Health
Map
Actions
Backend VM Group
us-west-spoke-2-instance-...
172.18.5.3
Google Virtual Machine
172.18.5.8
west-vpc-snet1-vm1(111095...
172.18.5.2
west-vpc-spoke2-firewall(...
Google Firewall
west-vpc-spoke2-router(75...
Google VPC Router
lb9-tcp-internal-lb(85338...
Google Load Balancer
172.18.5.0/28

4.1.3.6 Shared VPC

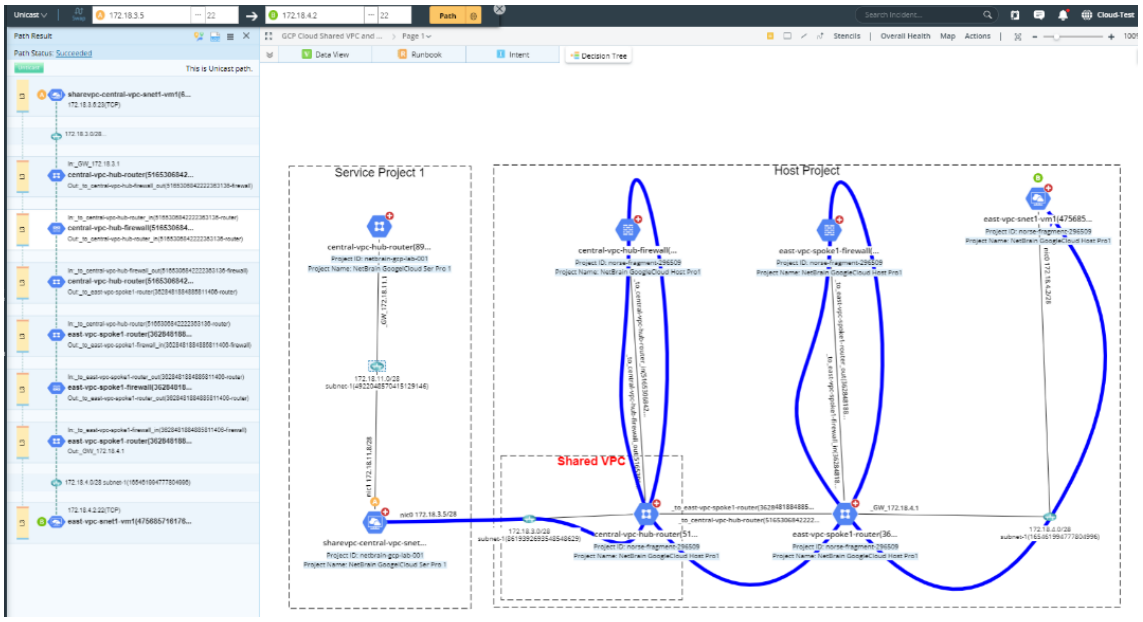
GCP Shared VPC allows an organization to connect resources from multiple projects to a common Virtual Private Cloud (VPC) network to communicate with each other securely and efficiently using internal IPs from that network.


Path Result
Path Status: Succeeded
This is Unicast path.
Data View
Runbook
Intent
Decision Tree
Service Project 1
Host Project
Shared VPC
central-vpc-hub-router(89...
sharevpc-central-vpc-snet...
central-vpc-hub-firewall(...
east-vpc-spoke1-firewall(...
east-vpc-snet1-vm1(475685...
central-vpc-hub-router(51...
east-vpc-spoke1-router(36...

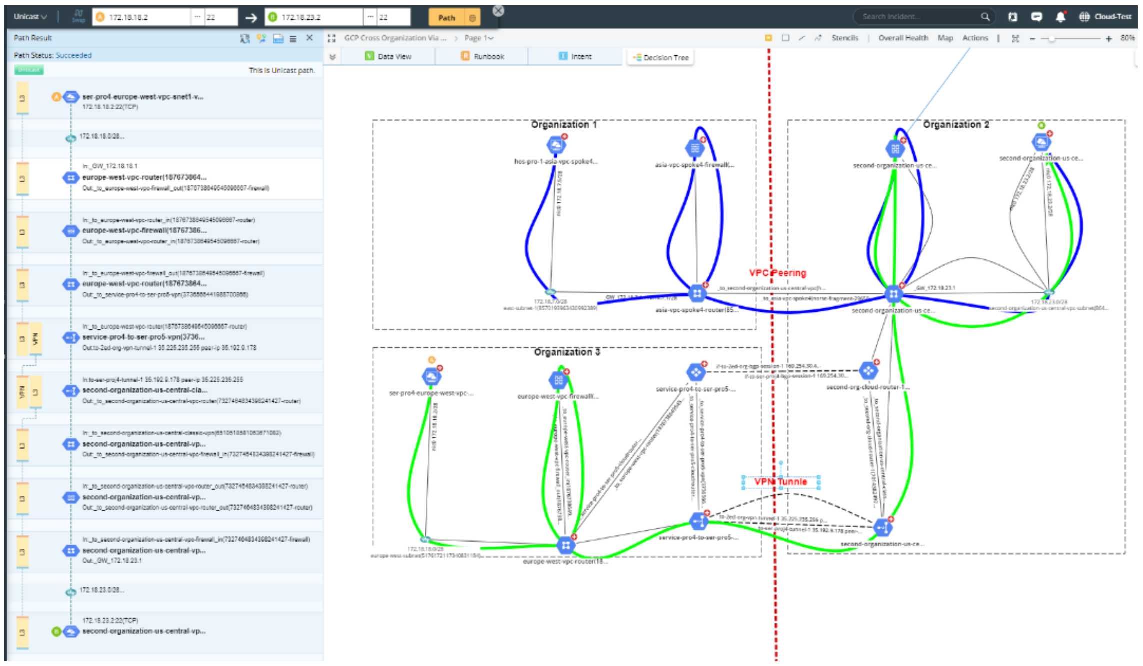
4.1.3.7 Cross Multiple Projects or Organizations

NetBrain supports visualizing the topology and path of the resources crossing multiple projects or organizations.

The following diagram demonstrates the path through Service Project 1 and Host Project.

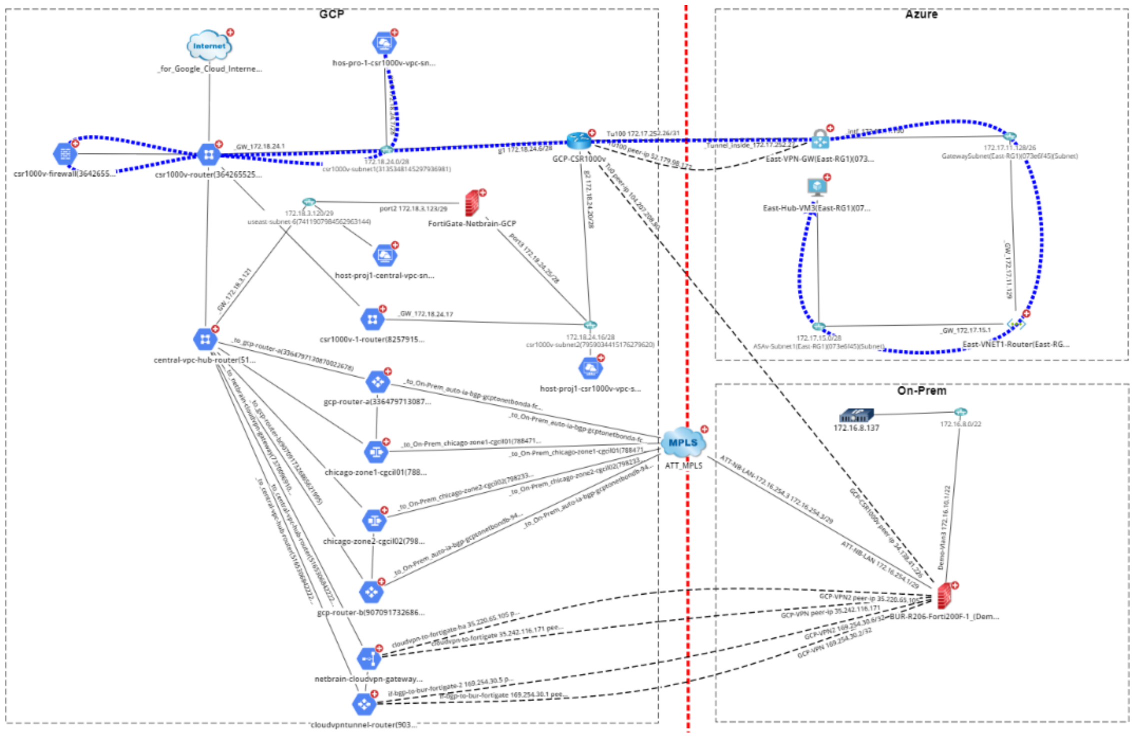


The following diagram demonstrates a path crossing two organizations via VPC Peering or VPN Tunnel.



4.1.3.8 Path across Multi-Cloud

NetBrain supports visualizing the topology and path across different public Clouds such as AWS, Azure, and GCP.

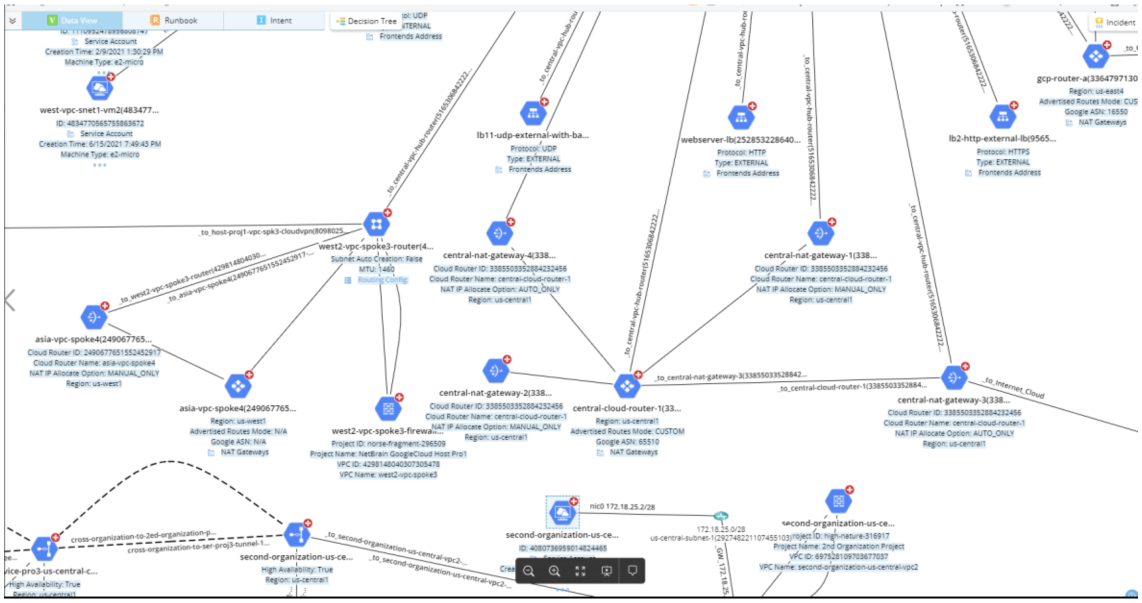

GCP
Azure
On-Prem
Internet
_for_Google_Cloud_Interne...
hos-pro-1-csr1000v-vpc-sn...
csr1000v-firewall(3642655...
csr1000v-router(364265525...
GCP-CSR1000v
FortiGate-Netbrain-GCP
host-proj1-central-vpc-sn...
csr1000v-1-router(8257915...
central-vpc-hub-router(51...
host-proj1-csr1000v-vpc-s...
gcp-router-a(336479713087...
chicago-zone1-cgcil01(788...
chicago-zone2-cgcil02(798...
gcp-router-b(907091732686...
netbrain-cloudvpn-gateway...
cloudvpntunnel-router(903...
MPLS
ATT_MPLS
East-VPN-GW(East-RG1)(073...
East-Hub-VM3(East-RG1)(07...
East-VNET1-Router(East-RG...
172.16.8.137
BUR-R206-Forti200F-1_(Dem...

4.1.4 SPOG with Cloud Native and 3rd Party Management Tools

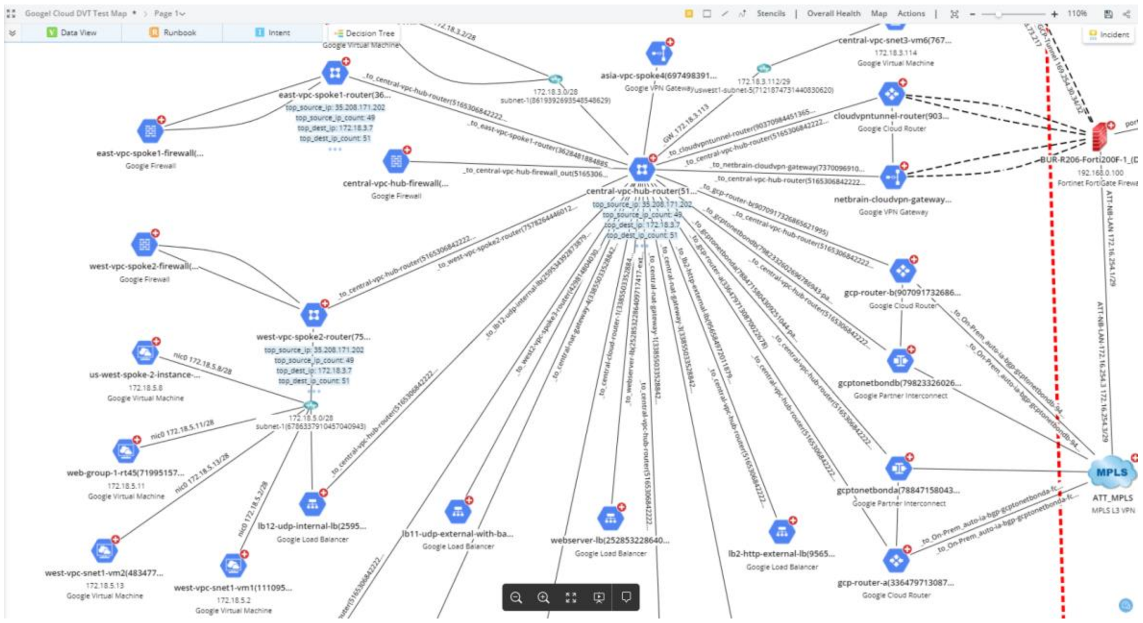
Data view (DVT) allows you to monitor various public data in the NetBrain environment. There are two kinds of data that the system can visualize:

- **Public Cloud Infrastructure Data**: the basic information of cloud operational status, routing/security, tag information, etc.
- **Cloud Monitoring Metrics**: the monitoring metrics from the cloud-native monitoring tools.

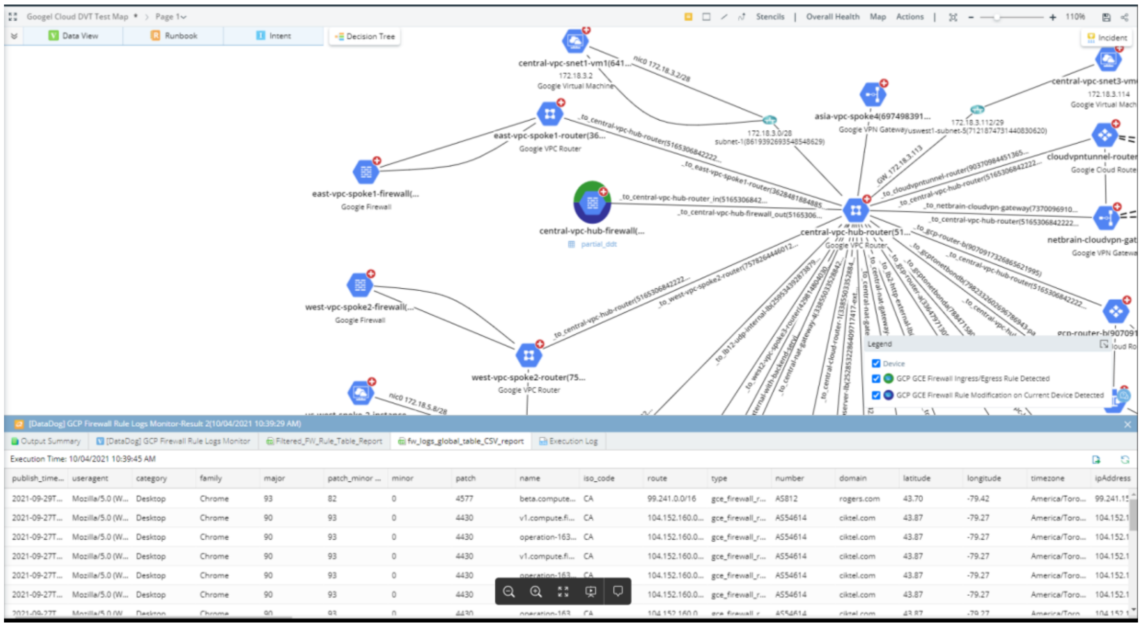
For example, the built-in **GPC basic info** DVT displays a GCP resource's basic information or properties. The different types of resources have different types of properties. For example, project ID, Project name, VPC ID, and VPC name for the Google Firewall; subnet auto-creation, MTU, and routing configuration for the Google VPC router.



Another DVT example, **VPC Flow Logs Analysis**, monitors the abnormal VPC traffic from specific IPs and regions. For example, if DDOS attacks the GCP Network, the source IP from a specific region will increase sharply.



Visualizing data from 3rd Party cloud management tools enables NetBrain dynamic map to render a complete view of your cloud infrastructures. The integration with 3rd Party cloud management tools is customizable, and NetBrain can integrate with any cloud management tool to suit your specific needs. An example is an integration with the Datadog: the system collects and analyzes the GCP firewall rule logs from Datadog and displays the results on the map. This integration can help find the history of a firewall rule and trace down any mistakenly deleted firewall rules.



[DataDog] GCP Firewall Rule Logs Monitor-Result 2(10/04/2021 10:39:29 AM)

Output Summary | [DataDog] GCP Firewall Rule Logs Monitor | Filtered_FW_Rule_Table_Report | fw_logs_global_table_CSV_report | Execution Log

Execution Time: 10/04/2021 10:39:45 AM

publish_time...	useragent	category	family	major	patch_minor ...	minor	patch	name	iso_code	route	type	number	domain	latitude	longitude	timezone	ipAddress
2021-09-29T...	Mozilla/5.0 (W...	Desktop	Chrome	93	82	0	4577	beta.compute...	CA	99.241.0.0/16	gce_firewall_r...	AS812	rogers.com	43.70	-79.42	America/Toro...	99.241.15
2021-09-27T...	Mozilla/5.0 (W...	Desktop	Chrome	90	93	0	4430	v1.compute.fi...	CA	104.152.160.0...	gce_firewall_r...	AS54614	ciktel.com	43.87	-79.27	America/Toro...	104.152.1
2021-09-27T...	Mozilla/5.0 (W...	Desktop	Chrome	90	93	0	4430	operation-163...	CA	104.152.160.0...	gce_firewall_r...	AS54614	ciktel.com	43.87	-79.27	America/Toro...	104.152.1
2021-09-27T...	Mozilla/5.0 (W...	Desktop	Chrome	90	93	0	4430	v1.compute.fi...	CA	104.152.160.0...	gce_firewall_r...	AS54614	ciktel.com	43.87	-79.27	America/Toro...	104.152.1
2021-09-27T...	Mozilla/5.0 (W...	Desktop	Chrome	90	93	0	4430	operation-163...	CA	104.152.160.0...	gce_firewall_r...	AS54614	ciktel.com	43.87	-79.27	America/Toro...	104.152.1
2021-09-27T...	Mozilla/5.0 (W...	Desktop	Chrome	90	93	0	4430	[illegible]	[illegible]	104.152.160.0...	gce_firewall_r...	AS54614	ciktel.com	43.87	-79.27	America/Toro...	104.152.1
2021-09-27T	Mozilla/5.0 (W	Desktop	Chrome	90	93	0	4430	operation-163	CA	104.152.160.0	gce_firewall_r	AS54614	ciktel.com	43.87	-79.27	America/Toro	104.152.1

4.2 IPv6 Network Support

V10.1 adds the support of the single-stack IPv6 network, including the discovery of IPv6 only network devices, L3/L2 topology, and mapping.

4.2.1 Discover Device based on IPv6 Address

In V10.1, the discovery functionalities have been improved to support the discovery of single-stack IPv6 devices, including:

- **Discovery via Seed Routers:** supports the input of both pure IPv6 address and mix of IPv4/IPv6 addresses

  Note: *This release does not support the neighbor walking of IPv6 address discovery.*

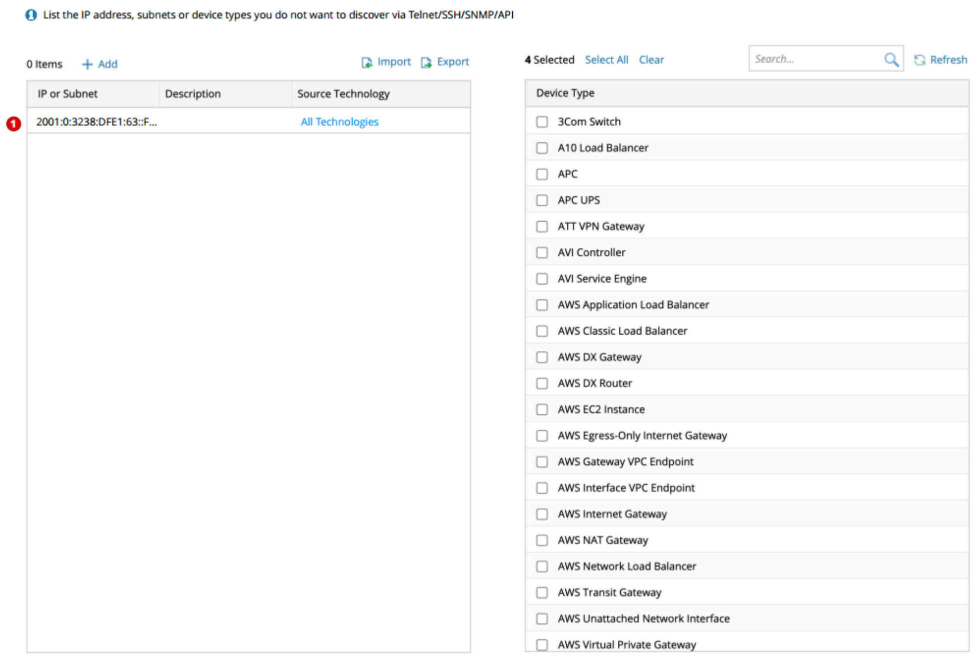
- **Scan IP Range:** supports the input of both single IPv6 address and range of IPv6 address.

  Note: *The IPv6 network segment must be specified by the mask, and the mask must be less than or equal to the 104-bit mask).*

Other discovery related improvements of IPv6 support includes:

1. **Support IPv6 Address in Do Not Scan Definition:**

   - Supports entering both IPv4 and IPv6 subnets into a Do not scan list when defining Do Not Scan based on Subnet.
   - The input IP address format (IPv4 or IPv6) can be automatically identified.
   - The input subnet mask range supports dynamic change.



List the IP address, subnets or device types you do not want to discover via Telnet/SSH/SNMP/API

0 Items + Add Import Export

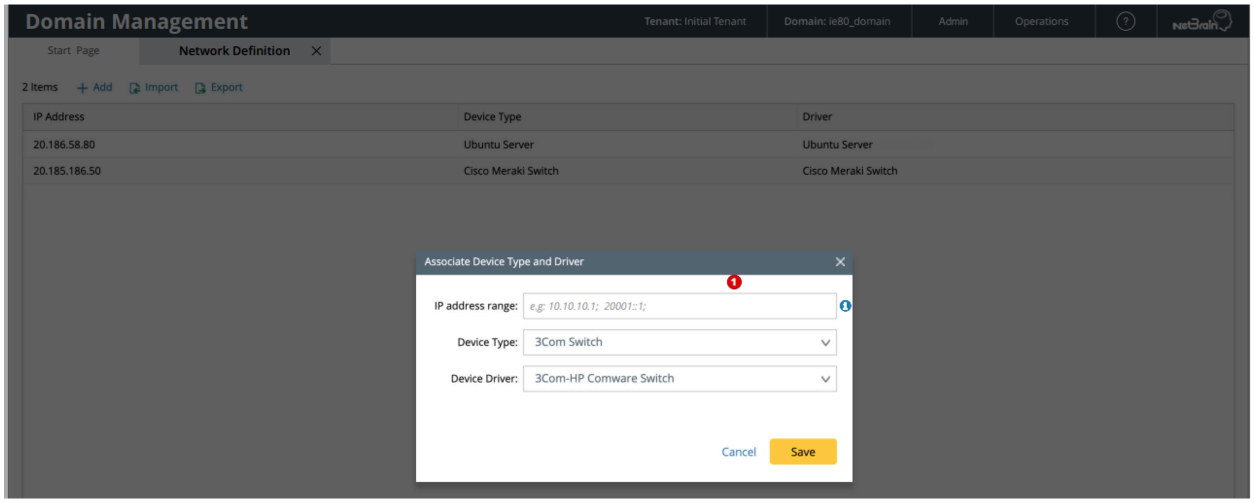
IP or Subnet	Description	Source Technology
2001:0:3238:DFE1:63::F...		All Technologies

4 Selected Select All Clear Search... Refresh

Device Type
3Com Switch
A10 Load Balancer
APC
APC UPS
ATT VPN Gateway
AVI Controller
AVI Service Engine
AWS Application Load Balancer
AWS Classic Load Balancer
AWS DX Gateway
AWS DX Router
AWS EC2 Instance
AWS Egress-Only Internet Gateway
AWS Gateway VPC Endpoint
AWS Interface VPC Endpoint
AWS Internet Gateway
AWS NAT Gateway
AWS Network Load Balancer
AWS Transit Gateway
AWS Unattached Network Interface
AWS Virtual Private Gateway

2. **Support IPv6 Address in Network Definition:** Network Definition allows users to manually define the device type and driver used by an IP address range when the device type and driver device cannot be accurately

identified through SNMP. The definition of IPv6 is supported in the IP address range, which is consistent with the IP range specs in Discovery.



IP Address	Device Type	Driver
20.186.58.80	Ubuntu Server	Ubuntu Server
20.185.186.50	Cisco Meraki Switch	Cisco Meraki Switch

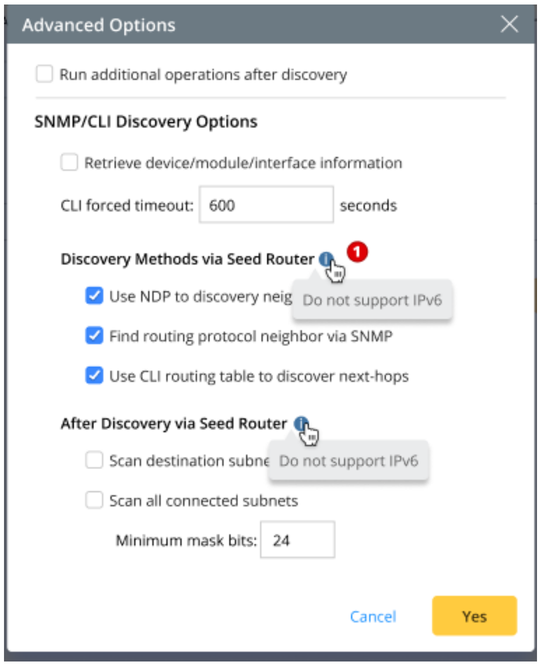
Associate Device Type and Driver

IP address range: e.g: 10.10.10.1; 20001::1;

Device Type: 3Com Switch

Device Driver: 3Com-HP Comware Switch

3. **Advanced Options for Discovery:** Some Advanced Options (neighbor walking) do not support IPv6 in this release.


Advanced Options

☐ Run additional operations after discovery

SNMP/CLI Discovery Options

☐ Retrieve device/module/interface information

CLI forced timeout: 600 seconds

Discovery Methods via Seed Router

☑ Use NDP to discovery neig Do not support IPv6

☑ Find routing protocol neighbor via SNMP

☑ Use CLI routing table to discover next-hops

After Discovery via Seed Router

☐ Scan destination subne Do not support IPv6

☐ Scan all connected subnets

Minimum mask bits: 24

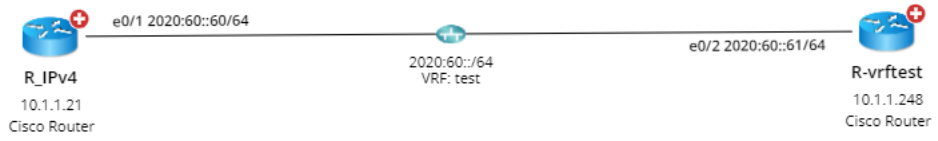
4. **Import IP List Template:** a download template file menu and a template file are provided for user reference.

4.2.2 Topology Improvements

IPv6 L3 topology has been improved, with additional support for the following network technologies:

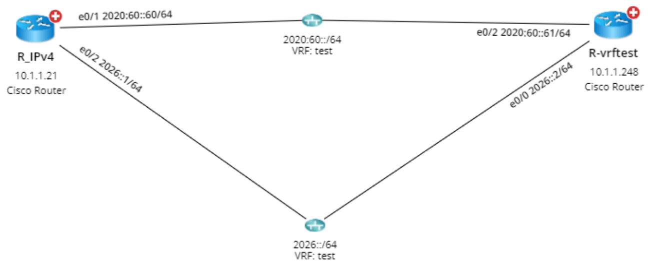
1. VRF

The VRF configured on the interface with an IPv6 address configuration will be parsed and applied to the IPv6 L3 topology calculation. When multiple interfaces are configured with the same IPv6 address prefix, and some of these interfaces are configured with the same VRF, a LAN media with the VRF name will be used to connect these interfaces.

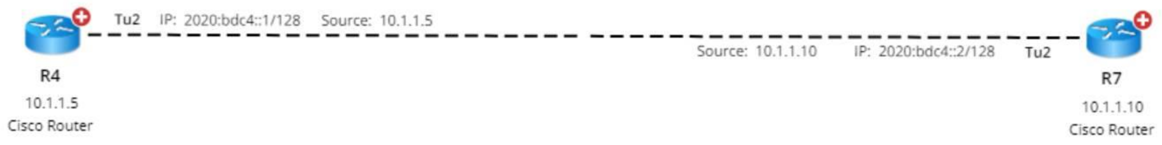


2. Secondary IP

When multiple IPv6 addresses are configured on the same interface, IPv6 L3 topologies will be created for these IPv6 addresses, respectively.

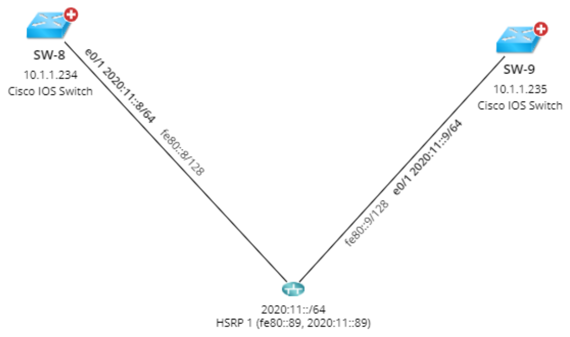


3. IPv6 over IPv4 GRE tunnel
   Support IPv6 over IPv4 GRE tunnel technology for IPv6 topology.


Tu2 IP: 2020:bdc4::1/128 Source: 10.1.1.5
Source: 10.1.1.10 IP: 2020:bdc4::2/128 Tu2
R4
10.1.1.5
Cisco Router
R7
10.1.1.10
Cisco Router

4. HSRP/VRRP/GLBP

If an IPv6 address is configured with First Hop Redundancy Protocols such as HSRP/VRRP/GLBP, the specific protocol name, group ID, and virtual IP address on the LAN media connecting these interfaces will be displayed.



4.2.3 More Built-in Data to Support IPv6

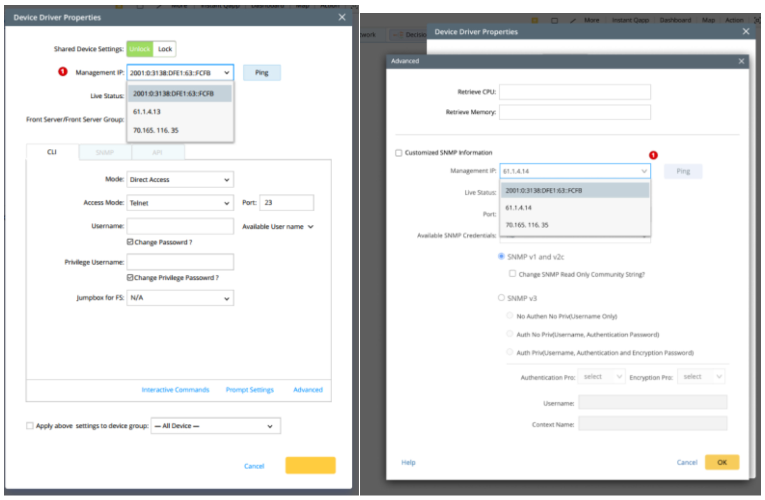
The following built-in data have been added to the NetBrain platform to facilitate the subsequent data analysis and better support the IPv6 related functionalities:

#	Data Table	Description	Improved Status
1	Route Table	Add a new NCT table: IPv6 Route Table.	Only support data retrieval via CLI

2	CDP Table	Add two more columns to the existing CDP table to retrieve Global IPv6 Address and Link Local Address.	Only support data retrieval via CLI
3	Neighbor Discover Table	Provide a separate IPv6 neighbor table for IPv6; the columns resolved by NCT are: Type, IPv6 Global Address, MAC address, Interface, Vlan, Private Vlan, Age, State, Time Left.	Only support data retrieval via CLI
4	Interface Information	Add the additional command of the IPv6 interface to analyze the IPv6 interface related information when parsing the config file.	Only support data retrieval via CLI

4.2.4 More Functions Supporting IPv6

1. **Share/Private Device Settings**: the device management IP supports IPv4 and IPv6.


Device Driver Properties

Shared Device Settings: Unlock Lock

Management IP: 2001:0:3138:DFE1:63::FCFB

2001:0:3138:DFE1:63::FCFB
61.1.4.13
70.165. 116. 35

Live Status:

Front Server/Front Server Group:

CLI | SNMP | API

Mode: Direct Access

Access Mode: Telnet Port: 23

Username: Available User name

☑ Change Passowrd ?

Privilege Username:

☑ Change Privilege Passowrd ?

Jumpbox for FS: N/A

Interactive Commands Prompt Settings Advanced

☐ Apply above settings to device group: — All Device —

Cancel

Advanced

Retrieve CPU:

Retrieve Memory:

☐ Customized SNMP Information

Management IP: 61.1.4.14 Ping

2001:0:3138:DFE1:63::FCFB
61.1.4.14
70.165. 116. 35

Live Status:

Port:

Available SNMP Credentials:

● SNMP v1 and v2c

☐ Change SNMP Read Only Community String?

○ SNMP v3

○ No Authen No Priv(Username Only)

○ Auth No Priv(Username, Authentication Password)

○ Auth Priv(Username, Authentication and Encryption Password)

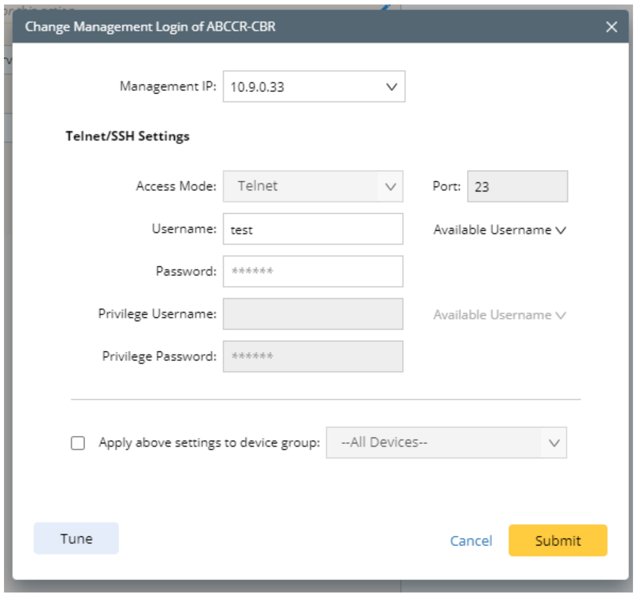
Authentication Pro: select Encryption Pro: select

Username:

Context Name:

Help Cancel OK

2. **Change Management Login Settings** now supports IPv4 and IPv6.


Change Management Login of ABCCR-CBR

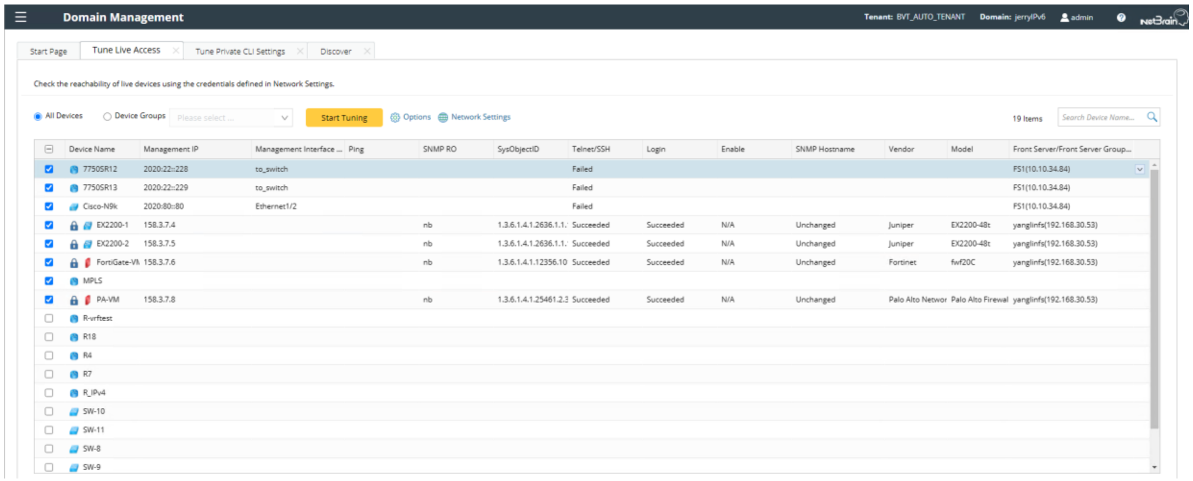
Management IP:	10.9.0.33		

Telnet/SSH Settings

Access Mode:	Telnet	Port:	23
Username:	test	Available Username	
Password:	******		
Privilege Username:		Available Username	
Privilege Password:	******		

☐ Apply above settings to device group: --All Devices--

3. **Batch Tune**: the management IP in both Tune Live Access Settings and Tune Private CLI Settings now supports IPv4 and IPv6. In addition, the management IP in the UI supports the display of IPv4/IPv6 address format.


Domain Management
Tenant: BVT_AUTO_TENANT Domain: jerryIPv6 admin
Start Page
Tune Live Access
Tune Private CLI Settings
Discover

Check the reachability of live devices using the credentials defined in Network Settings.

	Device Name	Management IP	Management Interface ...	Ping	SNMP RO	SysObjectID	Telnet/SSH	Login	Enable	SNMP Hostname	Vendor	Model	Front Server/Front Server Group...
☑	7750SR12	2020:22::228	to_switch				Failed						FS1(10.10.34.84)
☑	7750SR13	2020:22::229	to_switch				Failed						FS1(10.10.34.84)
☑	Cisco-N9k	2020:80::80	Ethernet1/2				Failed						FS1(10.10.34.84)
☑	EX2200-1	158.3.7.4			nb	1.3.6.1.4.1.2636.1.1.	Succeeded	Succeeded	N/A	Unchanged	Juniper	EX2200-48t	yanglinfs(192.168.30.53)
☑	EX2200-2	158.3.7.5			nb	1.3.6.1.4.1.2636.1.1.	Succeeded	Succeeded	N/A	Unchanged	Juniper	EX2200-48t	yanglinfs(192.168.30.53)
☑	FortiGate-VN	158.3.7.6			nb	1.3.6.1.4.1.12356.10	Succeeded	Succeeded	N/A	Unchanged	Fortinet	fwf20C	yanglinfs(192.168.30.53)
☑	MPLS												
☑	PA-VM	158.3.7.8			nb	1.3.6.1.4.1.25461.2.3	Succeeded	Succeeded	N/A	Unchanged	Palo Alto Networ	Palo Alto Firewal	yanglinfs(192.168.30.53)
☐	R-vrftest												
☐	R18												
☐	R4												
☐	R7												
☐	R_IPv4												
☐	SW-10												
☐	SW-11												
☐	SW-8												
☐	SW-9												

4. **Fine Tune**: the following functions of Fine Tune also support the IPv6 address format.

#	Function	Description
1	Unknown IP	Unknown IP supports IPv6 address format.
2	Discovered Device	Management IP supports IPv6 address format.
3	Missed Devices	Management IP supports IPv6 address format.
4	Discovered by SNMP	Management IP supports IPv6 address format.
5	Unknow SNMP SysObjectID	Management IP supports IPv6 address format.
6	Unclassified Network Devices	Management IP supports IPv6 address format.

5 Collaborative Troubleshooting Enhancements

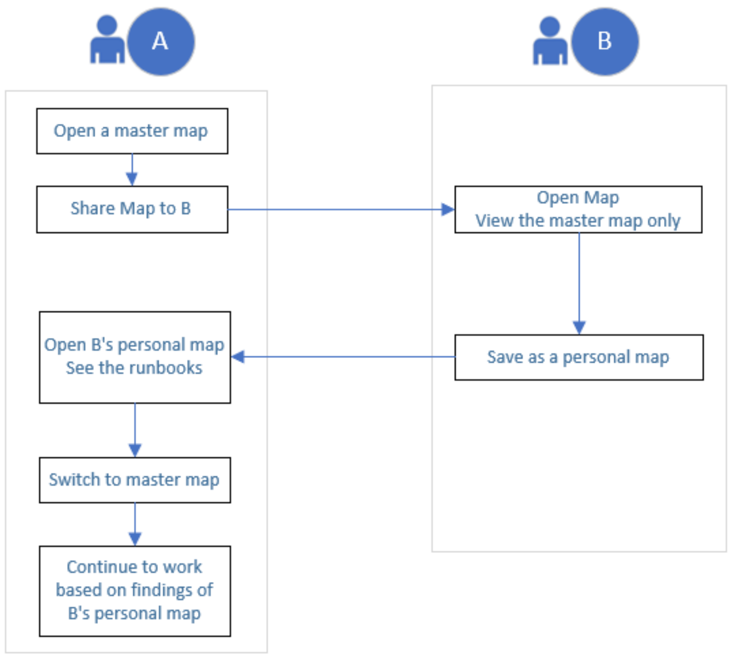
5.1 Personal Map Copy

In previous versions, one map could be edited by multiple users concurrently. However, only the map owner can save the changes on the map. Multiple users may operate simultaneously on a shared map such as the site map while troubleshooting the same problem. The limitation that only the owner can save and share the changes in a map can hinder team collaborations.

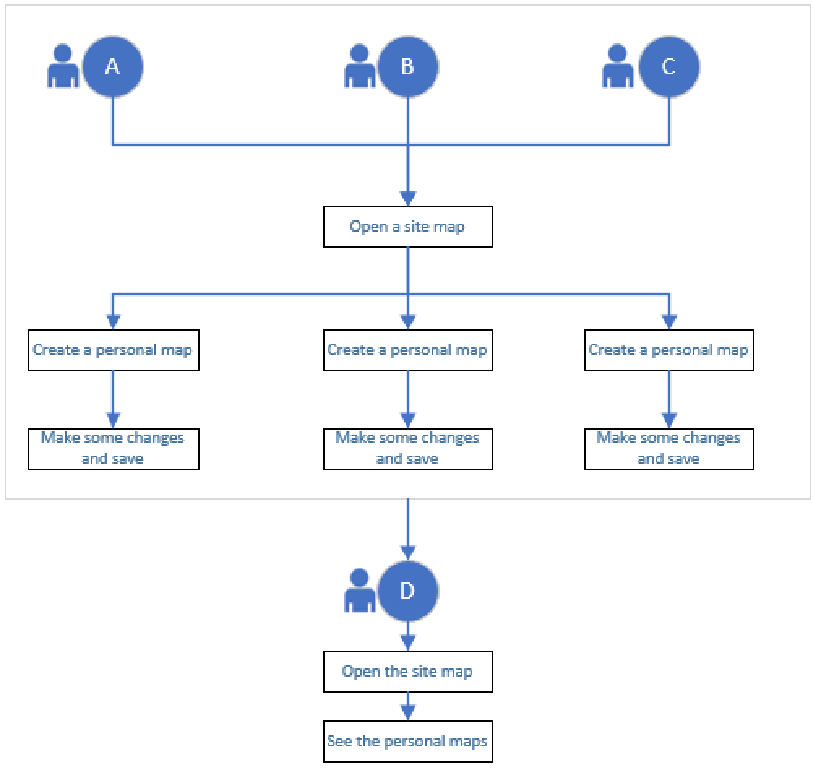
IE v10.1 introduces the **Personal Map Copy** function to facilitate better collaborations. Users can create a personal copy of a shared map (the master map) that they do not own. Then, users can save a personal copy and share their findings and changes with others without altering the master map.

Personal Copy can help two types of collaboration:

- Real-time collaboration: user B saves a map owned by user A as a personal copy and shares this personal copy with user A. User A can see any changes and actions user B makes in this personal copy.

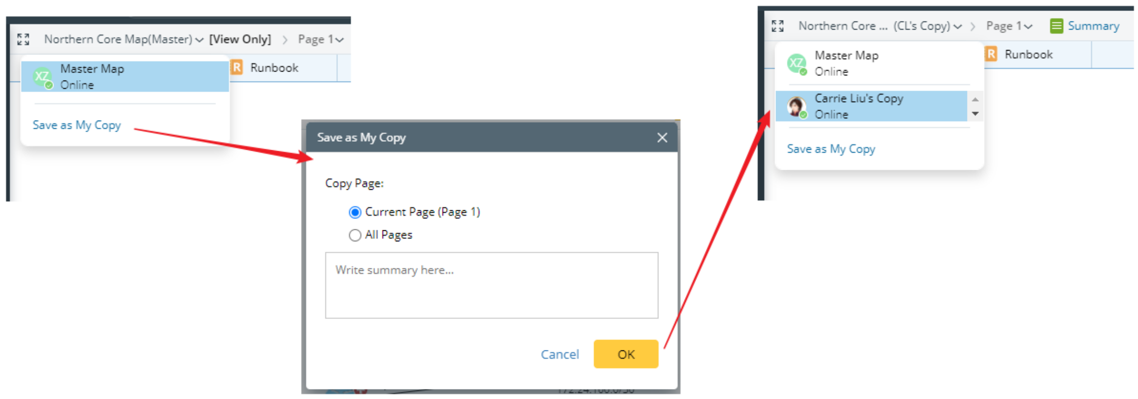


- Non-real-time collaboration: users create their personal copy maps from a shared site map.



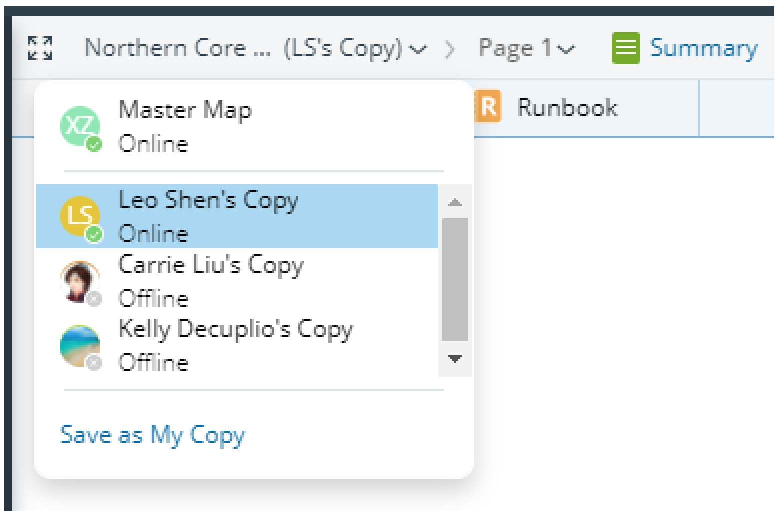
5.1.1 Create Personal Copy

Users can save any map as a personal copy with the **Save as My Copy** operation. He can select **Current Page** or **All Pages** for the personal copy. After the personal map copy is generated, changes can be made to this copy without impacting the master map.



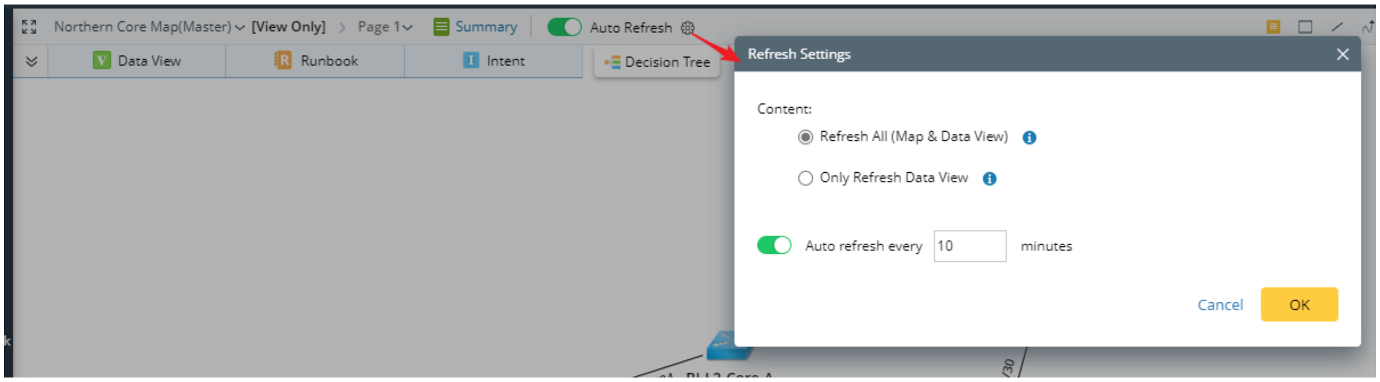
5.1.2 Switch Map Copy

Users can browse other users' copies of a master map. The personal copy shows the owner's initials (Avatar) and the status (whether the owner logins to the IE system).



5.1.3 Auto Refresh

Auto Refresh will be available to users when the current map (can be the master map or the personal copies owned by others) is opened in View-Only mode, i.e., the user has no editing right to this map.



Refresh Settings

Content:

Refresh All (Map & Data View) [✓]

Only Refresh Data View

Auto refresh every 10 minutes

Open a view-only map and enable **Auto Refresh.** Then, any changes by other users on the map will be automatically synchronized to the current view-only map. As a result, users will view the latest version of the map.

Users can configure the refresh settings:

- **Refresh All (Map & Data View)**

This option means reloading the latest saved map first and then reapplying the data view last selected in the map when refreshed.

- **Only Refresh Data View**

This option means reapplying the data view currently selected in the map when refreshed. For example, if the map owner drags and moves devices on a map to change the map layout or zoom in/out the map, the view-only map layout remains unchanged when refreshed.

5.2 Reference Map Enhancements

In the previous versions, users can create maps associated with NI, path, site, device group, and other types of objects. These maps are not standalone and can only be accessed from hosting objects. These maps cannot be shared across features, so users must copy the original map. These hidden maps can be divided into two types: Function Map, created with functions like Site, device group, NI, and Path; Reference Map, referenced by Path, NI, and member NIs of a NIC.

V10.1 provides two main improvements:

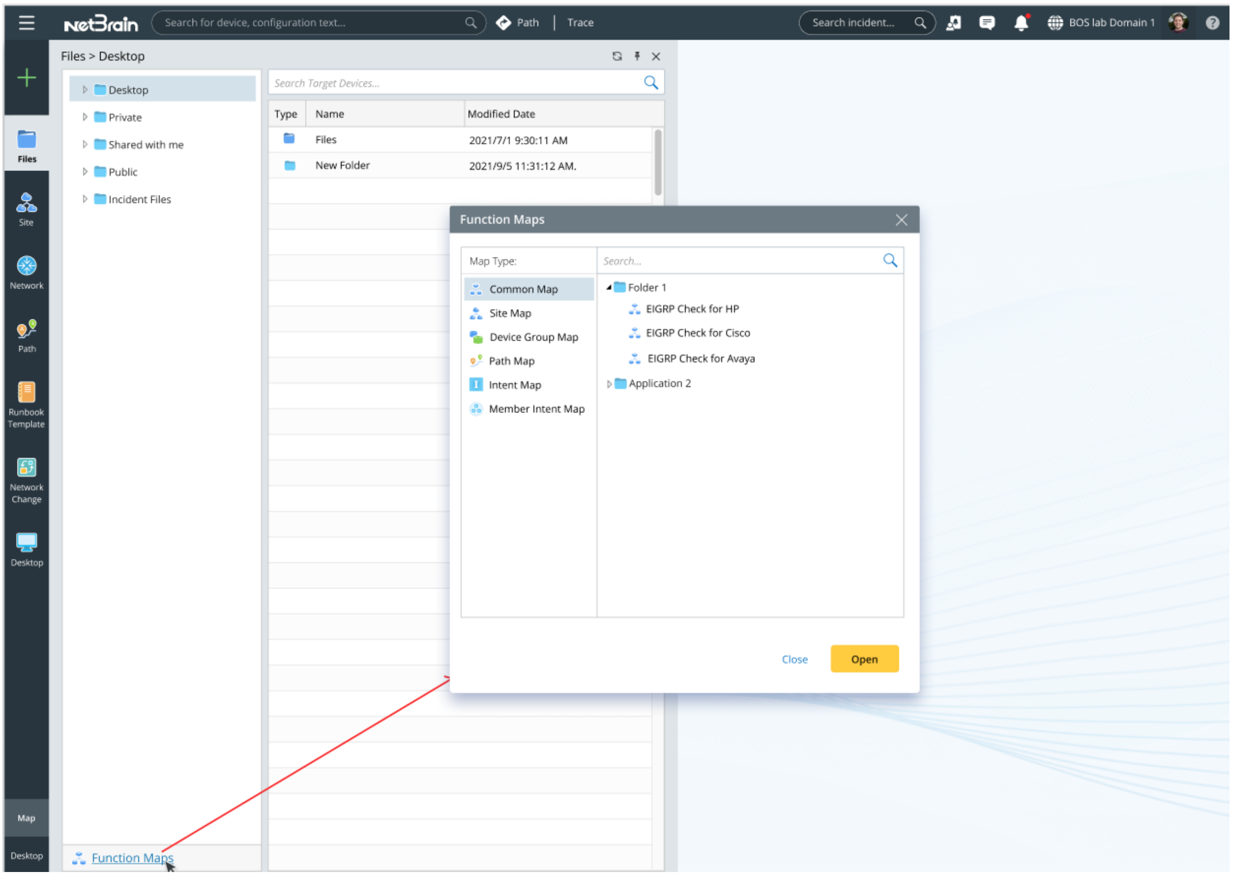
- Allow any common or function map as the reference map by Path, NI, and member NI.
- Browse all common maps and function maps in a central place.

The mainly function/reference map types are listed as follow:

Function	Map Type
Standalone Map	Common map
Site Map	Function Map
Device Group Map	Function Map
Network Intent Map	Function Map and Reference Map
Path Map	Function Map and Reference Map
Intent Cluster Member NI map	Function Map and Reference Map

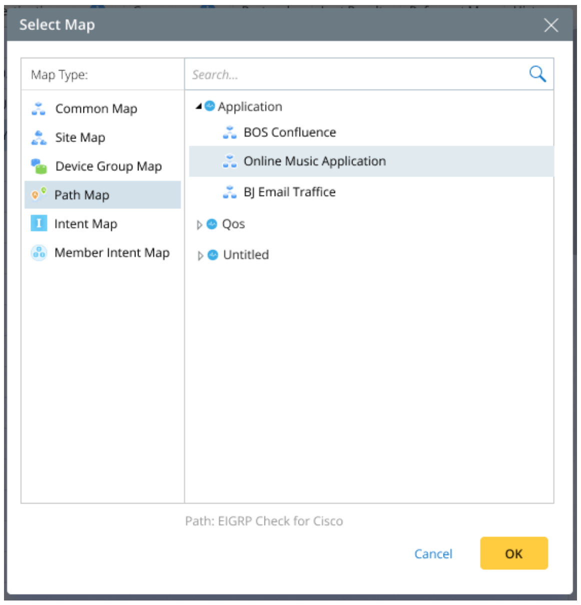
5.2.1 Function Maps Dialog

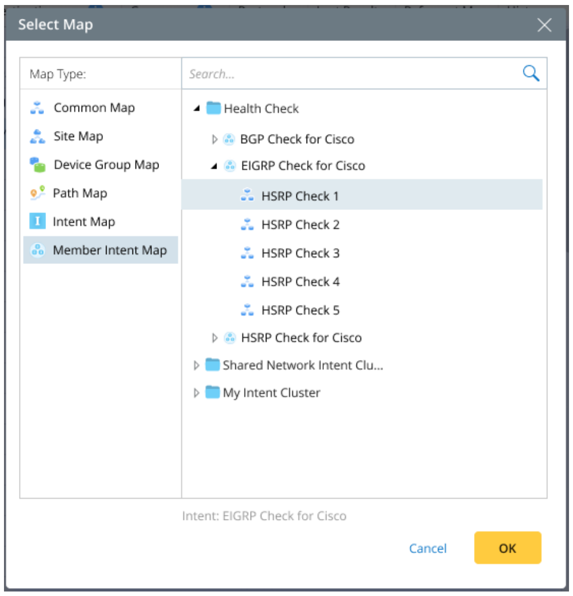
A central dialog, **Function Maps**, is provided for users to open all common, function, and reference maps.



Select a map type in the left pane, and all maps for this type will be displayed in the right pane. The maps are organized in the folder structure for the common, device group, and NI maps and the site tree for the site maps.

The path maps are displayed under the Applications, and the intent maps of NIC's NI members are under each NIC.





The system will automatically create the map if a user selects a device group map or a site map not generated yet.

5.2.2 Set Reference Map

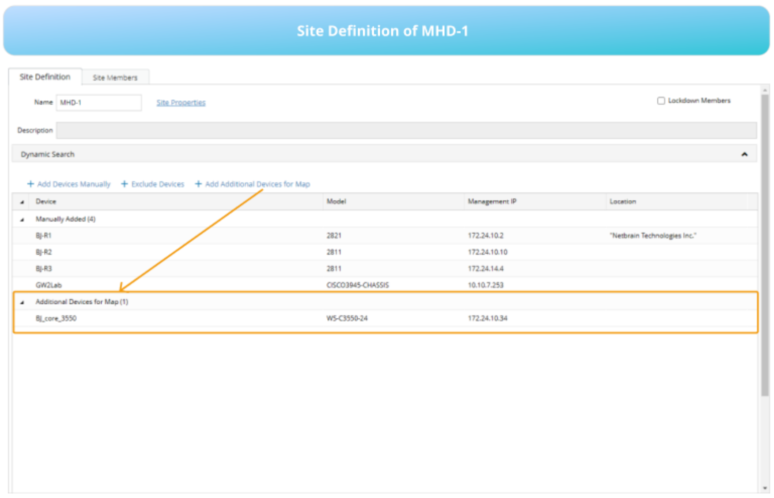
V10.1 makes two enhancements for the user to set a reference map for the path and NI:

- The **function maps** dialog is used to select a map for the reference map of Path, NI, member NIs of NIC, and Incent map.
- In earlier versions, when a map is set to be a reference map, the system makes a hard copy of the master map, and these two maps are totally independent and need to be maintained separately. V10.1 provides an option to allow a reference map is associated with the master map instead of a hard copy.

5.3 Site Map Enhancements

A site is often connected with other sites, and it will be helpful to show the detailed border link information in a site map. Currently, only site member devices are drawn on a site map.

In v10.1, users can add more devices to a site map, such as the linked neighbor devices. A feature, ***Add Additional Devices for Map***, is added in the site definition and devices. These additional devices will not be removed from the site map when the system benchmark task automatically updates site maps.

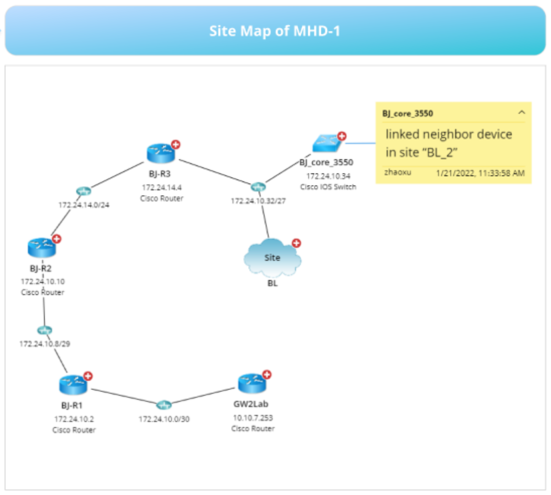


Name: MHD-1

Description:

☐ Lockdown Members

Device	Model	Management IP	Location
Manually Added (4)			
BJ-R1	2821	172.24.10.2	"Netbrain Technologies Inc."
BJ-R2	2811	172.24.10.10	
BJ-R3	2811	172.24.14.4	
GW2Lab	CISCO3945-CHASSIS	10.10.7.253	
Additional Devices for Map (1)			
BJ_core_3550	WS-C3550-24	172.24.10.34	



BJ_core_3550
linked neighbor device in site "BL_2"
zhaoxu 1/21/2022, 11:33:58 AM

Additional devices for a map only affect the mapping functions, and all other functions will not consider these devices as site member devices) such as inventory reports, searches, etc.

5.4 Smart CLI for MAC OS

The Smart CLI for Mac OS is added, with the same functions as the Smart CLI for Windows.

Using Smart CLI, network engineers can intelligently analyze the CLI output, automatically document the troubleshooting activities, and effectively collaborate with a team of co-workers.

6 Other Enhancements

6.1 KC and Auto Update Enhancements

With the newest enhancements in Knowledge Cloud (KC), delivering platform resources to customers requires less manual work. Additionally, In V10.1, more resources are included in platform resources, such as Open Driver and platform validation rules. Thus, customers can leverage more up-to-date resources than before.

The platform resources that can be updated automatically includes:

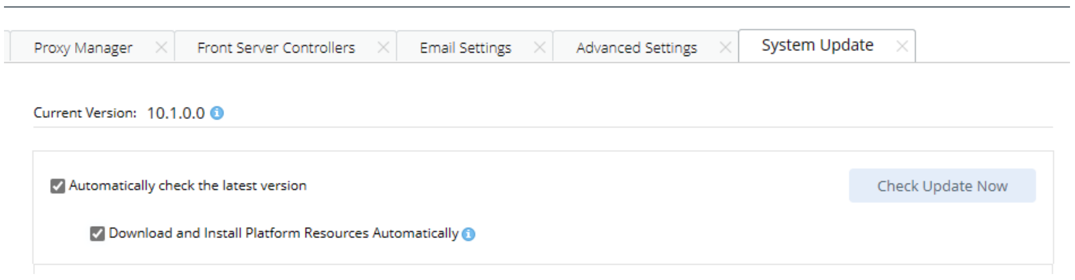
- Resources to create and maintain automation assets, such as FIT, Visual Parser, Guidebook, etc.
- Resources to verify/fix data accuracy or maintain the system's essential functions, such as Open Driver, platform validation rules, system driver, device type, etc.

6.1.1 Enhancements in KC

- Online update resources

In previous versions, KC has no direct connection with the NetBrain system. So NetBrain system administrator must download the software package from NetBrain Customer Portal and manually upload it to the system. In V10.1, the IE system will be connected to KC through License Server, allowing resources to be downloaded and installed to NetBrain IE and reducing the manual work of downloading and uploading the software package.

- Enable auto-download and auto-install resources
  Users can decide whether to download and install the resources automatically.



Current Version: 10.1.0.0

☑ Automatically check the latest version

☑ Download and Install Platform Resources Automatically

- Users can discard the uploaded version for framework components, even if the update has been scheduled.

6.1.2 Full Resources Updates

By enabling the **Download and Install Platform Resources Automatically** option, full resources will be updated, i.e., all types of platform resources will be downloaded. The full resources update is incremental, i.e., only new platform resources will be downloaded.

6.1.3 The List of Resources

The platform resources that can be auto updated and downloaded via KC:

Resource Name		
Qapp	**GDR Properties**	**Golden Baseline Dynamic Analysis Logic**
Gapp	**Tech Spec**	**Interface Type (Interface Name Translation)**
Runbook Template	**Media Type**	**Default Date View Template**
Data View Template	**API Plugin**	**Object Tree Template**
Parser Library	**SPOG URL**	**NGSystem. Technology License Definition**
Driver	**Visual Space**	**Multi Source Mapping**
Device Type	**Vendor Model Table**	**Domain system settings**
Device Icon & Picture	**Global Python Scripts (Built-in)**	**Cloud type**
Topology Link Type (IPv4, IPv6 etc.)	**Variable Mapping & Global Variable**	**FIT (Feature Intent Template)**

Generic Schema (Including Generic SSchema Icon)	**Platform Plugin**	**Full CLI Command**
CLI Command Template	**Beta Driver**	**PV File**
Open Driver	**/**	**/**

6.2 License Enhancements

In V10.0, the License Model is divided into three sub-modules: Foundation, Network, and Function. The Foundation Module is for the traditional network devices, while the Network Module can be further divided into various emerging network technologies. Each Network Module has its own independent license counting method. The license of network modules cannot be converted in V10.0, which is inconvenient for the customers migrating from the traditional network to the new technology such as the public cloud.

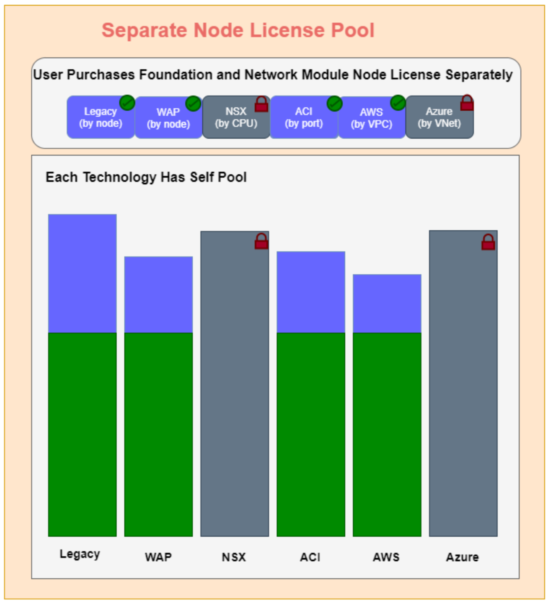
V10.1 introduces the following improvements:

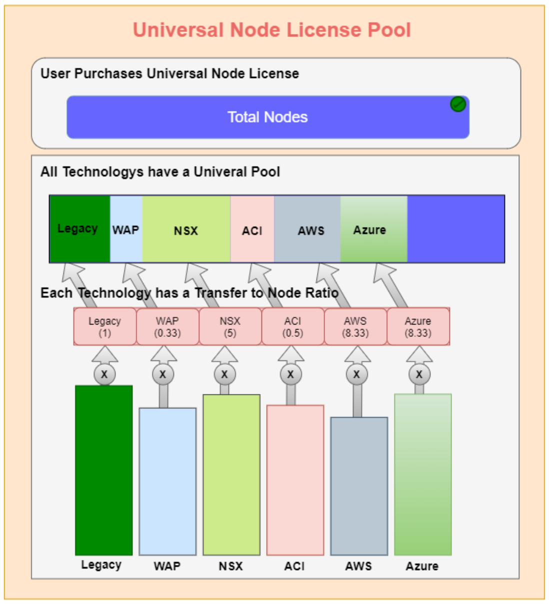
- Provide a new Universal Node License, allowing users to convert between Foundation and Network modules according to a defined ratio. The Universal Node License will coexist with the Separate Module License.
- Improve the IBA license, allowing users to use the corresponding functions up to many times without purchasing IBA. This enhancement can speed up the adoption of PDAS.

6.2.1 Universal Node License vs. Separate Module License

The Universal Node License will coexist with the Separate Module License, although one system can only have one license model.

- Separate Pool License Model: This license model in 10.0a remains unchanged.
- Universal Node License Model: The customer does not need the separate Network Modules under this model. Instead, Network Modules will be converted to the universal nodes according to a specific ratio.


Separate Node License Pool
User Purchases Foundation and Network Module Node License Separately
Legacy (by node)
WAP (by node)
NSX (by CPU)
ACI (by port)
AWS (by VPC)
Azure (by VNet)
Each Technology Has Self Pool
Legacy
WAP
NSX
ACI
AWS
Azure


Universal Node License Pool
User Purchases Universal Node License
Total Nodes
All Technologys have a Univeral Pool
Legacy
WAP
NSX
ACI
AWS
Azure
Each Technology has a Transfer to Node Ratio
Legacy (1)
WAP (0.33)
NSX (5)
ACI (0.5)
AWS (8.33)
Azure (8.33)
X
Legacy
WAP
NSX
ACI
AWS
Azure

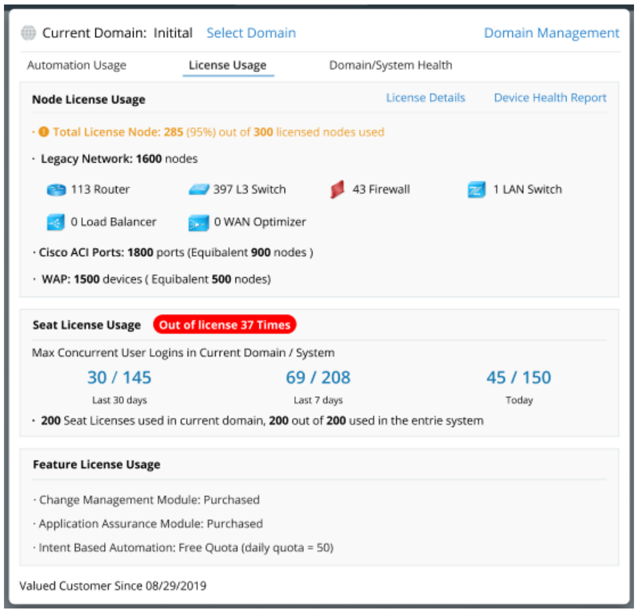
6.2.2 IBA License Adjustments

To speed up PDAS adoptions, V10.1 adjusts the IBA license as follows:

- There are no restrictions on NI/NIC creation and viewing of NI/NIC history.
- NI/NIC execution (both manually and triggered via API, including running NI inside API stub) will be metered daily per domain. The system will stop executing NI/NIC after a certain quota, and the customer needs to buy the IBA license to run NI/NIC without any limitation.
- Users can switch to weekly usage metering instead of daily metering.
- Preventative automation does not need the IBA license, including 1) Define AM probe; 2) View Trend chart; 3) View IBA dashboard.

6.2.3 Function Improvements for IE

V10.1 supports switching from Separate Pool License mode to Universal Pool License mode, and the functions and the relevant UI are adjusted accordingly, including the system license page, license assignment page, license usage page.


Current Domain: Initital Select Domain Domain Management

Automation Usage License Usage Domain/System Health

Node License Usage License Details Device Health Report

- **Total License Node: 285** (95%) out of **300** licensed nodes used
- **Legacy Network: 1600** nodes

113 Router 397 L3 Switch 43 Firewall 1 LAN Switch

0 Load Balancer 0 WAN Optimizer

- **Cisco ACI Ports: 1800** ports (Equibalent **900** nodes)
- **WAP: 1500** devices (Equibalent **500** nodes)

Seat License Usage Out of license 37 Times

Max Concurrent User Logins in Current Domain / System

30 / 145	69 / 208	45 / 150
Last 30 days	Last 7 days	Today

- **200** Seat Licenses used in current domain, **200** out of **200** used in the entrie system

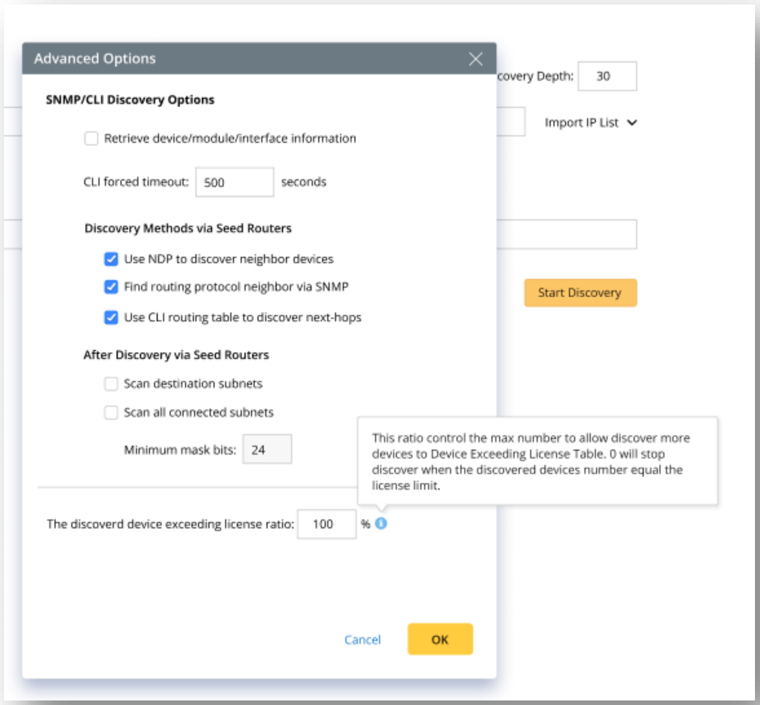
Feature License Usage

- Change Management Module: Purchased
- Application Assurance Module: Purchased
- Intent Based Automation: Free Quota (daily quota = 50)

Valued Customer Since 08/29/2019

6.2.4 Discover Devices Exceeding Licensed Nodes

Users often do not know how many devices are in their network and do not purchase enough nodes. V10.1 allows users to discover more devices than the licensed nodes. The devices exceeding the license are listed separately and cannot be mapped and participate in other functions. The user can configure how many more devices are discovered in the Advanced Setting of Discovery.


Advanced Options

SNMP/CLI Discovery Options

☐ Retrieve device/module/interface information

CLI forced timeout: 500 seconds

Discovery Methods via Seed Routers

☑ Use NDP to discover neighbor devices

☑ Find routing protocol neighbor via SNMP

☑ Use CLI routing table to discover next-hops

After Discovery via Seed Routers

☐ Scan destination subnets

☐ Scan all connected subnets

Minimum mask bits: 24

The discoverd device exceeding license ratio: 100 %

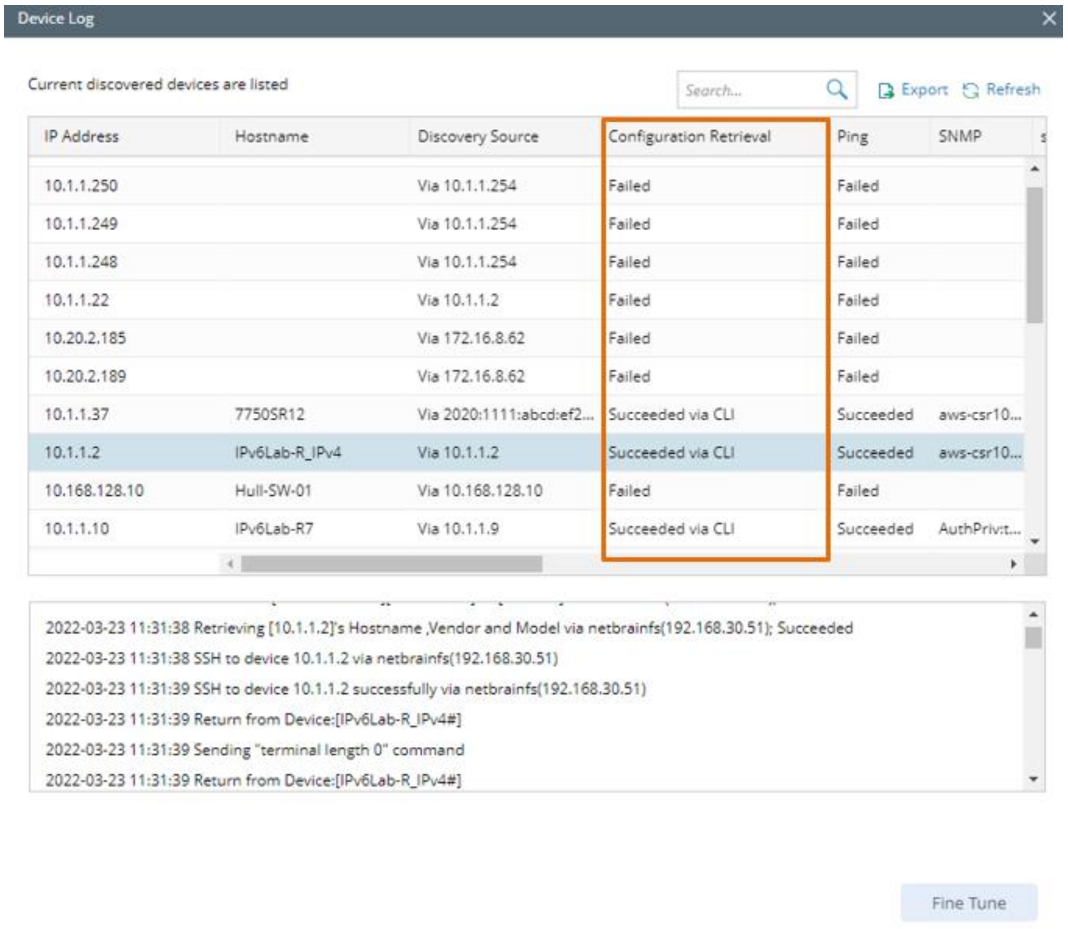
This ratio control the max number to allow discover more devices to Device Exceeding License Table. 0 will stop discover when the discovered devices number equal the license limit.

6.3 Discovery, Benchmark, Live Access, and Fine-Tune Enhancements

6.3.1 Discovery Enhancements

V10.1 made the following improvements on discovery:

- When the system fails to retrieve the device configurations via CLI, the Configuration column is displayed as ***Failed*** instead of ***Succeeded via SNMP*** in the device log.



Device Log

Current discovered devices are listed

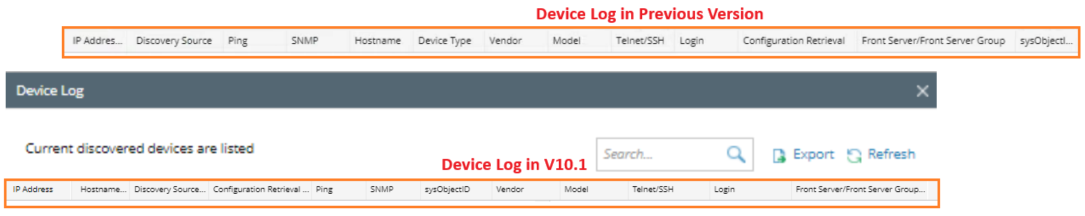
Search... Export Refresh

IP Address	Hostname	Discovery Source	Configuration Retrieval	Ping	SNMP
10.1.1.250		Via 10.1.1.254	Failed	Failed	
10.1.1.249		Via 10.1.1.254	Failed	Failed	
10.1.1.248		Via 10.1.1.254	Failed	Failed	
10.1.1.22		Via 10.1.1.2	Failed	Failed	
10.20.2.185		Via 172.16.8.62	Failed	Failed	
10.20.2.189		Via 172.16.8.62	Failed	Failed	
10.1.1.37	7750SR12	Via 2020:1111:abcd:ef2...	Succeeded via CLI	Succeeded	aws-csr10...
10.1.1.2	IPv6Lab-R_IPv4	Via 10.1.1.2	Succeeded via CLI	Succeeded	aws-csr10...
10.168.128.10	Hull-SW-01	Via 10.168.128.10	Failed	Failed	
10.1.1.10	IPv6Lab-R7	Via 10.1.1.9	Succeeded via CLI	Succeeded	AuthPriv:t...

```
2022-03-23 11:31:38 Retrieving [10.1.1.2]'s Hostname ,Vendor and Model via netbrainfs(192.168.30.51); Succeeded
2022-03-23 11:31:38 SSH to device 10.1.1.2 via netbrainfs(192.168.30.51)
2022-03-23 11:31:39 SSH to device 10.1.1.2 successfully via netbrainfs(192.168.30.51)
2022-03-23 11:31:39 Return from Device:[IPv6Lab-R_IPv4#]
2022-03-23 11:31:39 Sending "terminal length 0" command
2022-03-23 11:31:39 Return from Device:[IPv6Lab-R_IPv4#]
```

Fine Tune

- The columns in the device log are reorganized for better display.



Device Log in Previous Version

IP Addres...	Discovery Source	Ping	SNMP	Hostname	Device Type	Vendor	Model	Telnet/SSH	Login	Configuration Retrieval	Front Server/Front Server Group	sysObjectI...

Device Log

Current discovered devices are listed

Search... Export Refresh

Device Log in V10.1

IP Address	Hostname...	Discovery Source...	Configuration Retrieval ...	Ping	SNMP	sysObjectID	Vendor	Model	Telnet/SSH	Login	Front Server/Front Server Group...

- In previous versions, the devices that fail in Ping or SNMP in **Scan IP Range** will not be listed in the device log. In v10.1, the devices the user enters in the discovery input box will be listed in the device log even when they fail with Ping or SNMP. The neighbor devices discovered by the system will not be listed in the device log of **Scan IP Range** if they fail with Ping or SNMP.

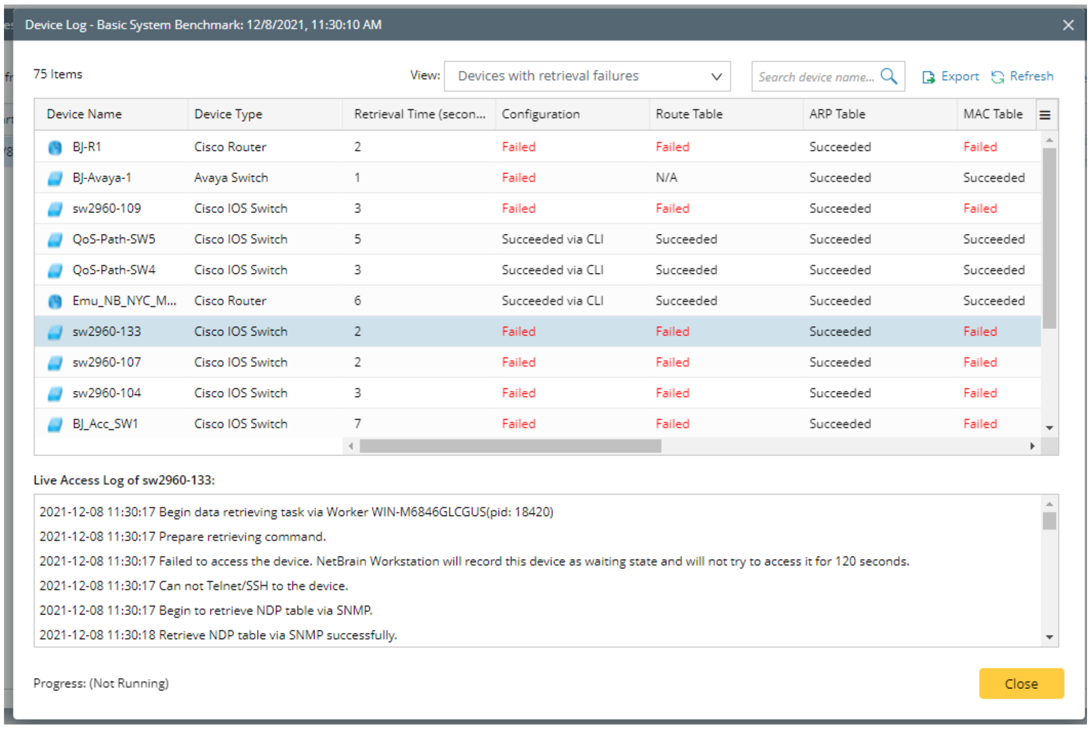
- To maintain the management IP of the current devices in the domain, for neighbor devices successfully discovered in **Discover via Seed Routers**, the system will first check if those devices are in the current domain; if they are in the domain, their management IP will not change. This logic does not apply to seed devices: the management IP of the seed device will be updated if the discovery succeeds and the management IP is not locked.

- Discovery via API can take hours to complete. In previous versions, the discovery logs are loaded only when the discovery task is finished. In v10.1, the system will update the discovery logs every 10 minutes to provide users with the latest discovery progress.

6.3.2 Benchmark Enhancements

V10.1 made two enhancements on benchmark:

- When the system fails to retrieve the device configurations via CLI, the Configuration column is displayed as ***Failed*** instead of ***Succeeded via SNMP*** in the device log.


Device Log - Basic System Benchmark: 12/8/2021, 11:30:10 AM

75 Items — View: Devices with retrieval failures

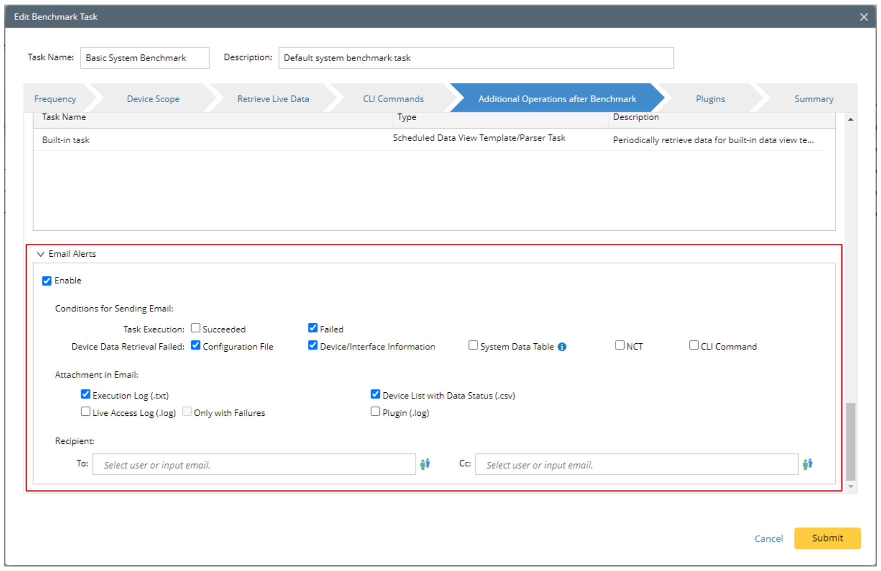
Device Name	Device Type	Retrieval Time (secon...	Configuration	Route Table	ARP Table	MAC Table
BJ-R1	Cisco Router	2	Failed	Failed	Succeeded	Failed
BJ-Avaya-1	Avaya Switch	1	Failed	N/A	Succeeded	Succeeded
sw2960-109	Cisco IOS Switch	3	Failed	Failed	Succeeded	Failed
QoS-Path-SW5	Cisco IOS Switch	5	Succeeded via CLI	Succeeded	Succeeded	Succeeded
QoS-Path-SW4	Cisco IOS Switch	3	Succeeded via CLI	Succeeded	Succeeded	Succeeded
Emu_NB_NYC_M...	Cisco Router	6	Succeeded via CLI	Succeeded	Succeeded	Succeeded
sw2960-133	Cisco IOS Switch	2	Failed	Failed	Succeeded	Failed
sw2960-107	Cisco IOS Switch	2	Failed	Failed	Succeeded	Failed
sw2960-104	Cisco IOS Switch	3	Failed	Failed	Succeeded	Failed
BJ_Acc_SW1	Cisco IOS Switch	7	Failed	Failed	Succeeded	Failed

Live Access Log of sw2960-133:

```
2021-12-08 11:30:17 Begin data retrieving task via Worker WIN-M6846GLCGUS(pid: 18420)
2021-12-08 11:30:17 Prepare retrieving command.
2021-12-08 11:30:17 Failed to access the device. NetBrain Workstation will record this device as waiting state and will not try to access it for 120 seconds.
2021-12-08 11:30:17 Can not Telnet/SSH to the device.
2021-12-08 11:30:17 Begin to retrieve NDP table via SNMP.
2021-12-08 11:30:18 Retrieve NDP table via SNMP successfully.
```

Progress: (Not Running)

- Add more settings to the **Email Alerts** section in the Benchmark definition interface to specify conditions to send the email and attached documents.


Edit Benchmark Task

Task Name: Basic System Benchmark   Description: Default system benchmark task

Task Name	Type	Description
Built-in task	Scheduled Data View Template/Parser Task	Periodically retrieve data for built-in data view te...

Email Alerts

☑ Enable

Conditions for Sending Email:

Task Execution: ☐ Succeeded ☑ Failed

Device Data Retrieval Failed: ☑ Configuration File ☑ Device/Interface Information ☐ System Data Table ☐ NCT ☐ CLI Command

Attachment in Email:

☑ Execution Log (.txt) ☑ Device List with Data Status (.csv)

☐ Live Access Log (.log) ☐ Only with Failures ☐ Plugin (.log)

Recipient:

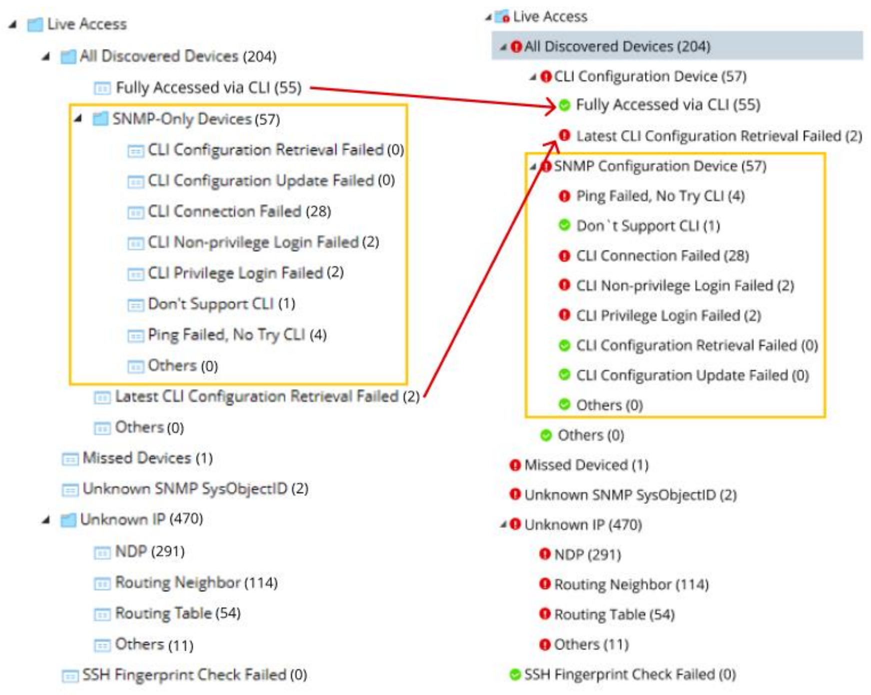
To: Select user or input email.   Cc: Select user or input email.

6.3.3 Fine Tune Enhancements

Discovery and Benchmark are foundations for building the network model, accessing live network devices, and retrieving the device data periodically. Both functions can generate the device accessibility and data integrity report. However, only the Discovery function generates the report in previous versions, and the Benchmark function does not. V10.1 improves the Benchmark function to create the report so that users can find the device accessibility issues to make the schedule benchmark tasks work well.

V10.1 also makes several usability improvements in the Fine Tune:

- Adjusts the report organization to adapt the data source extension and the logic improvements.
- Tell users intuitively which device accessibility report has alerts through the alert status icon.



Live Access
All Discovered Devices (204)
Fully Accessed via CLI (55)
SNMP-Only Devices (57)
CLI Configuration Retrieval Failed (0)
CLI Configuration Update Failed (0)
CLI Connection Failed (28)
CLI Non-privilege Login Failed (2)
CLI Privilege Login Failed (2)
Don't Support CLI (1)
Ping Failed, No Try CLI (4)
Others (0)
Latest CLI Configuration Retrieval Failed (2)
Others (0)
Missed Devices (1)
Unknown SNMP SysObjectID (2)
Unknown IP (470)
NDP (291)
Routing Neighbor (114)
Routing Table (54)
Others (11)
SSH Fingerprint Check Failed (0)
Live Access
All Discovered Devices (204)
CLI Configuration Device (57)
Fully Accessed via CLI (55)
Latest CLI Configuration Retrieval Failed (2)
SNMP Configuration Device (57)
Ping Failed, No Try CLI (4)
Don`t Support CLI (1)
CLI Connection Failed (28)
CLI Non-privilege Login Failed (2)
CLI Privilege Login Failed (2)
CLI Configuration Retrieval Failed (0)
CLI Configuration Update Failed (0)
Others (0)
Others (0)
Missed Deviced (1)
Unknown SNMP SysObjectID (2)
Unknown IP (470)
NDP (291)
Routing Neighbor (114)
Routing Table (54)
Others (11)
SSH Fingerprint Check Failed (0)

6.3.4 Front Server Group

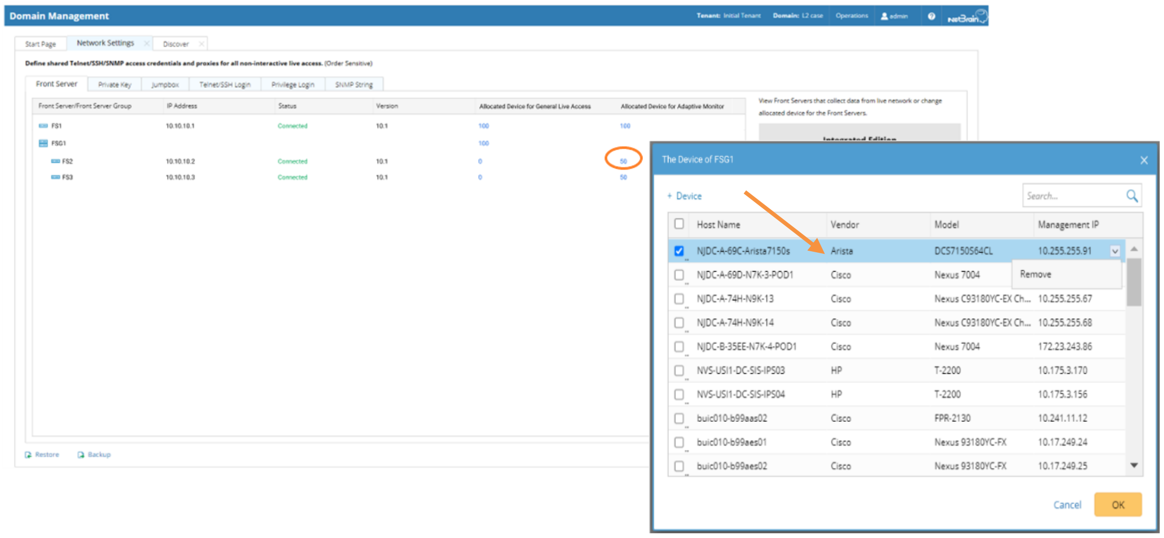
Front Server Group was removed in v10.0 when the data storage system was redesigned for Preventive Automation. V10.1 added it back to support the high availability function for Front Servers.

For Preventive Automation, the load balance function of the Front Server Group is not available since the data is stored on Front Server. Users need to manually migrate the data while switching the device from one Front Server to another. For all other functions to use the live access, such as discovery and benchmark, Front Server Group provides the load balance function: if one Front Server is down, the other available Front Server will be used.

Both Front Server and Front Server Groups can be selected in shared device settings. If a Front Server Group is selected for devices, NetBrain will equally assign those devices to Front Servers under the Front Server Group for Preventive Automation.

Users can manage the devices under a Front Server or Front Server Group by adding or removing devices from the managed device list.

- If devices are added to a Front Server Group, the Front Server Group is used for device live access, and those devices will be evenly assigned to Front Servers of the Front Server Group for Preventive Automation.
- If devices are removed from the Front Server Group or Front Server, they will also be removed from the managed device list for Preventive Automation. Similarly, if devices are removed from the Preventive Automation list, they will also be removed from the managed device list for device live access.



Define shared Telnet/SSH/SNMP access credentials and proxies for all non-interactive live access. (Order Sensitive)

Front Server/Front Server Group	IP Address	Status	Version	Allocated Device for General Live Access	Allocated Device for Adaptive Monitor
FS1	10.10.10.1	Connected	10.1	100	100
FSG1				100	
FS2	10.10.10.2	Connected	10.1	0	50
FS3	10.10.10.3	Connected	10.1	0	50

View Front Servers that collect data from live network or change allocated device for the Front Servers.

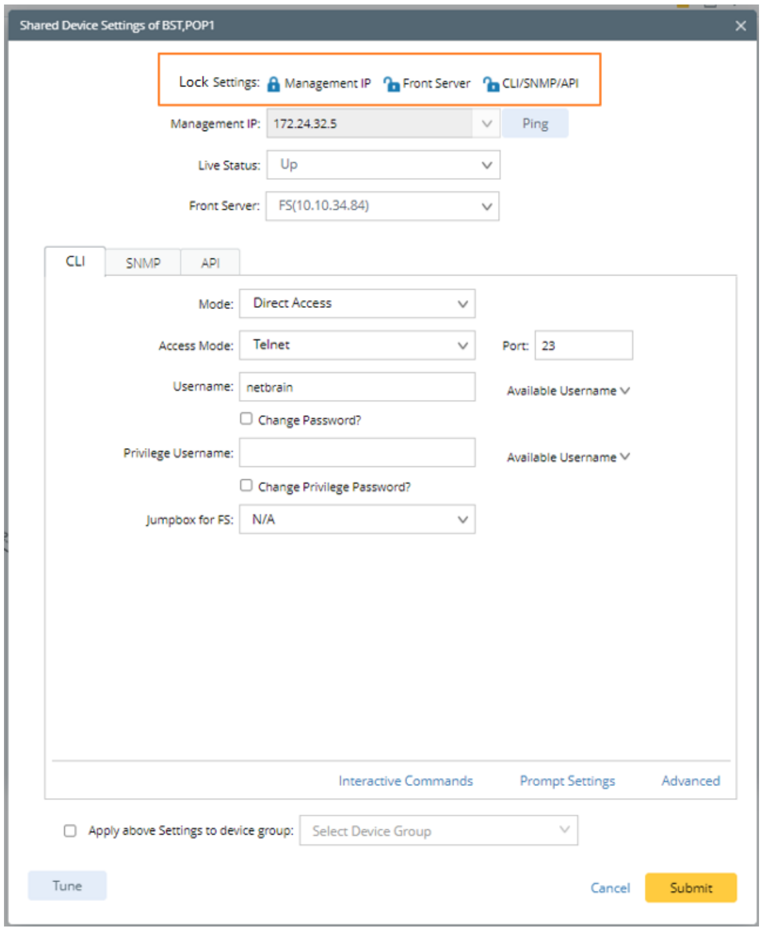
The Device of FSG1

	Host Name	Vendor	Model	Management IP
✓	NJDC-A-69C-Arista7150s	Arista	DCS7150S64CL	10.255.255.91
	NJDC-A-69D-N7K-3-POD1	Cisco	Nexus 7004	
	NJDC-A-74H-N9K-13	Cisco	Nexus C93180YC-EX Ch...	10.255.255.67
	NJDC-A-74H-N9K-14	Cisco	Nexus C93180YC-EX Ch...	10.255.255.68
	NJDC-B-35EE-N7K-4-POD1	Cisco	Nexus 7004	172.23.243.86
	NVS-USI1-DC-SIS-IPS03	HP	T-2200	10.175.3.170
	NVS-USI1-DC-SIS-IPS04	HP	T-2200	10.175.3.156
	buic010-b99aas02	Cisco	FPR-2130	10.241.11.12
	buic010-b99aes01	Cisco	Nexus 93180YC-FX	10.17.249.24
	buic010-b99aes02	Cisco	Nexus 93180YC-FX	10.17.249.25

The deployment method for the Front Server Group is the same as the deployment in v8.x. When the system is upgraded from v8.x to v10.1, the settings related to Front Server and the Front Server Group will be inherited kept.

6.3.5 Lock Only One Setting in Device Setting

In Shared Device Settings, users can lock/unlock only one setting: **Management IP, Front Server,** or **CLI/SNMP/API**.


Shared Device Settings of BST,POP1

Lock Settings: Management IP, Front Server, CLI/SNMP/API

Management IP: 172.24.32.5

Live Status: Up

Front Server: FS(10.10.34.84)

CLI tab:

Mode: Direct Access

Access Mode: Telnet

Port: 23

Username: netbrain

Privilege Username:

Jumpbox for FS: N/A

Correspondingly, the users can lock one of these settings on the **Fune Tune**.

6.4 Installation Pre-check Tool

The installation and upgrading of the system require the correct configurations of the servers. If these requirements are not met, installation/upgrade failures can happen. Or the installed/upgraded IE cannot run smoothly afterward.

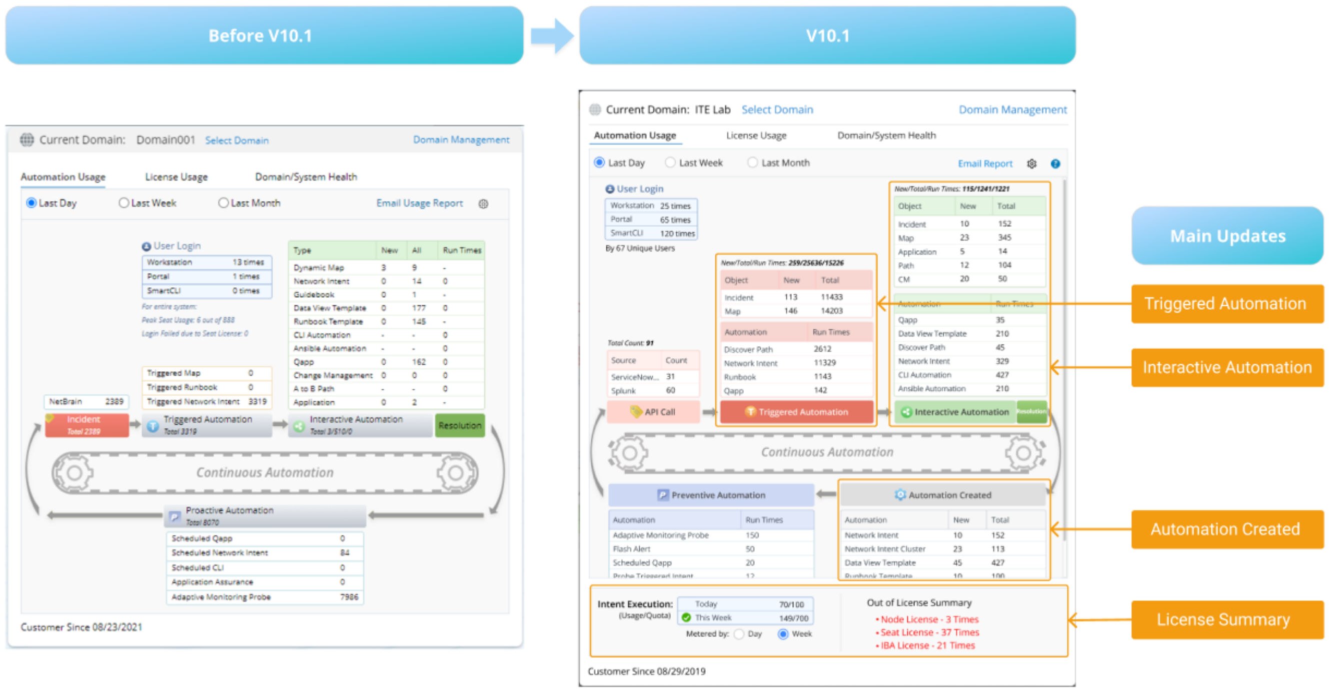
To enhance the success rate and simplify the process of installation/upgrade, v10.1 provides a pre-check tool to check users' system readiness to install/upgrade and generate a report to help users prepare appropriately for the installation/upgrade.

The tool supports all-in-two, distributed, and HA deployment and upgrading from 7.1x, 8.x, and 10.x to 10.1.

The command line based pre-check tool can check both Windows and Linux servers and report if these servers can meet the installation/upgrade requirements. If a requirement is not satisfied, more details may be provided. In addition, the report includes a summary listing all errors and warning messages.

6.5 Automation Usage Pane Enhancements

Automation Usage Pane provides the metric of the different types of automation, such as how many NIs are created and how many times NIs are executed. V10.1 optimizes the algorithm to calculate the metrics accurately and displays more metrics, as illustrated in the following diagram:



Before V10.1:

Workstation	13 times
Portal	1 times
SmartCLI	0 times

For entire system:
Peak Seat Usage: 6 out of 888
Login Failed due to Seat License: 0

Type	New	All	Run Times
Dynamic Map	3	9	-
Network Intent	0	14	0
Guidebook	0	1	-
Data View Template	0	177	0
Runbook Template	0	145	-
CLI Automation	-	-	0
Ansible Automation	-	-	0
Qapp	0	162	0
Change Management	0	0	0
A to B Path	-	-	0
Application	0	2	-

NetBrain	2389

Triggered Map	0
Triggered Runbook	0
Triggered Network Intent	3319

Scheduled Qapp	0
Scheduled Network Intent	84
Scheduled CLI	0
Application Assurance	0
Adaptive Monitoring Probe	7986

Customer Since 08/23/2021

V10.1:

Workstation	25 times
Portal	65 times
SmartCLI	120 times

By 67 Unique Users

Total Count: 91

Source	Count
ServiceNow...	31
Splunk	60

New/Total/Run Times: 259/25636/15226

Object	New	Total
Incident	113	11433
Map	146	14203

Automation	Run Times
Discover Path	2612
Network Intent	11329
Runbook	1143
Qapp	142

New/Total/Run Times: 115/1241/1221

Object	New	Total
Incident	10	152
Map	23	345
Application	5	14
Path	12	104
CM	20	50

Automation	Run Times
Qapp	35
Data View Template	210
Discover Path	45
Network Intent	329
CLI Automation	427
Ansible Automation	210

Automation	Run Times
Adaptive Monitoring Probe	150
Flash Alert	50
Scheduled Qapp	20
Probe Triggered Intent	12

Automation	New	Total
Network Intent	10	152
Network Intent Cluster	23	113
Data View Template	45	427
Runbook Template	10	100

Intent Execution: (Usage/Quota)

Today	70/100
✓ This Week	149/700

Metered by: Day [✓ Week]

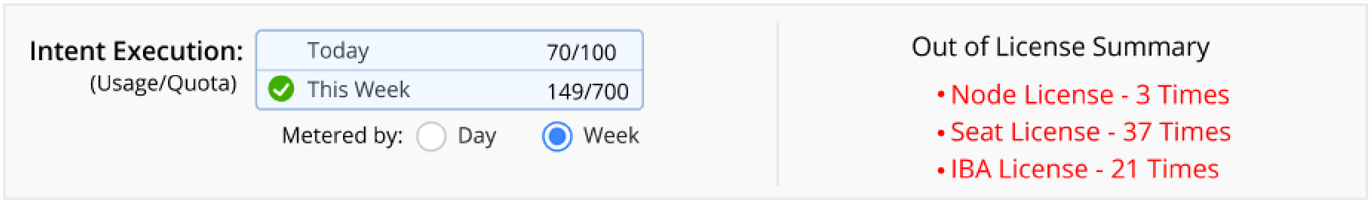
Out of License Summary
- Node License - 3 Times
- Seat License - 37 Times
- IBA License - 21 Times

Customer Since 08/29/2019

- Triggered automation and interactive automation are displayed separately.
- The Objects (e.g., Incident, Map, etc.) are classified into new and total categories.
- The run times of the automation, such as Discover Path, Network Intent, etc., are counted.

6.5.1 Intent Quota and License Summary

Users can view the Intent Execution usage and quota when the IBA license is not purchased. Users can switch the meter between week and day with Metered by. The number of times when the node license, seat license, and IBA license are exceeded is also displayed.



Intent Execution: (Usage/Quota)

Today	70/100
✓ This Week	149/700

Metered by: Day [✓ Week]

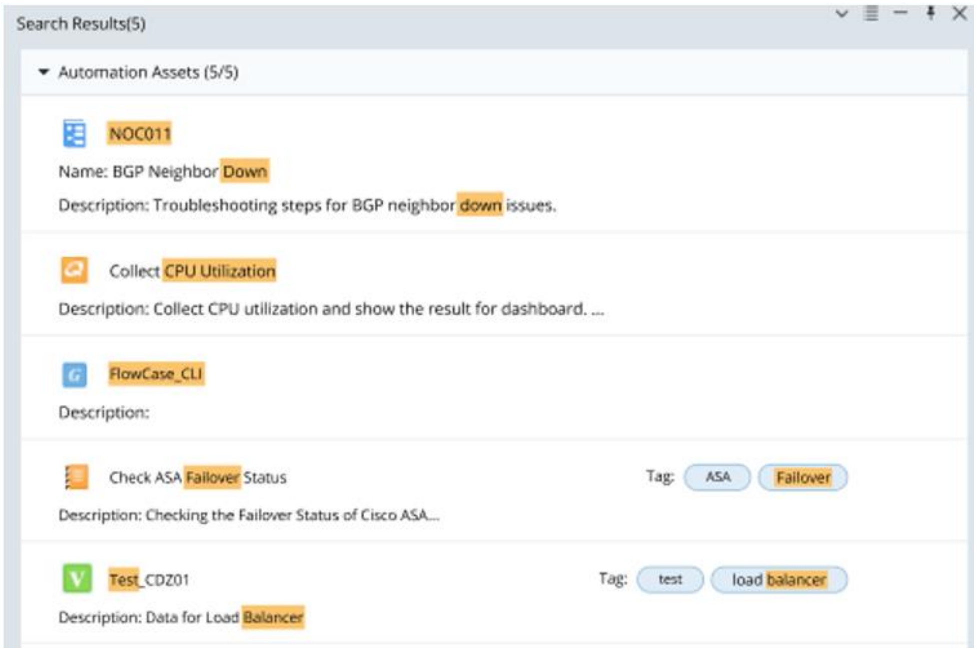
Out of License Summary

- Node License - 3 Times
- Seat License - 37 Times
- IBA License - 21 Times

6.6 Search Enhancements

V10.1 expands the search functions to more objects, including:

- the automation assets such as Qapp, Gapp, Runbook, and DVT. The results are listed under the category ***Automation Assets***.


Search Results(5)

Automation Assets (5/5)

NOC011

Name: BGP Neighbor Down

Description: Troubleshooting steps for BGP neighbor down issues.

Collect CPU Utilization

Description: Collect CPU utilization and show the result for dashboard. ...

FlowCase_CLI

Description:

Check ASA Failover Status Tag: ASA Failover

Description: Checking the Failover Status of Cisco ASA...

Test_CDZ01 Tag: test load balancer

Description: Data for Load Balancer

- NCT (Network Control Tables): two NCT are searchable: Virtual server table and NAT table. All columns in the table can be searched, and the system will search only the current baseline data.

6.7 New REST APIs

V10.1 adds the following REST APIs:

- TAF Management
  - ✓ Create IT System Data Model API
  - ✓ Auto Trigger API
  - ✓ Get Trigger Result API
  - ✓ Get Trigger Diagnosis Definition API
  - ✓ Manually Trigger API
  - ✓ Update Incident Message API
  - ✓ Get Temporary Incident Portal Access Token API
  - ✓ Verify User Permission API
- API Server Management:
  - ✓ Get All API Servers Configured in Domain API

- AWS Account Management
    - ✓ Add AWS Account API
    - ✓ Delete AWS Account API
    - ✓ Get AWS Account API
    - ✓ Get AWS Accounts API
    - ✓ Patch AWS Account API
- Authentication and Authorization
    - ✓ Get All Accessible Domains API
    - ✓ Get All Accessible Tenants API
- Device Group Management
    - ✓ Add Device Group
    - ✓ Get Device Group API
    - ✓ Get Group Devices API

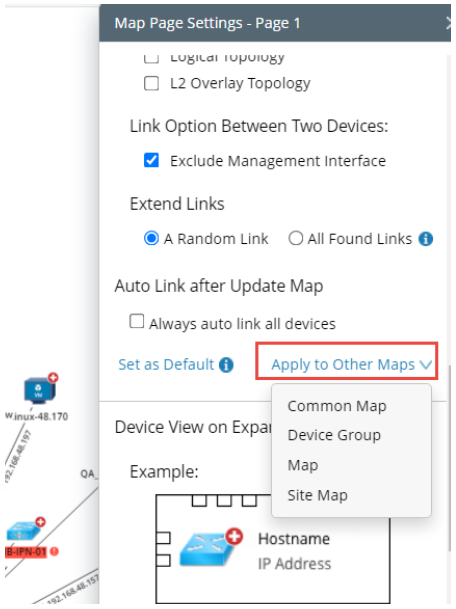
Refer to https://github.com/NetBrainAPI/NetBrain-REST-API-R10.1 for details of new and updated APIs.

6.8 Other Improvements and Adjustments

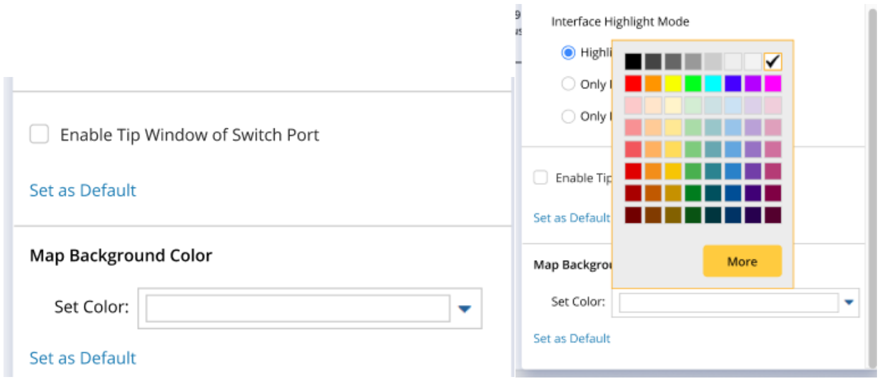
6.8.1 Other Map Improvements

Besides the personal map copy and reference map, v10.1 has the following map related improvements:

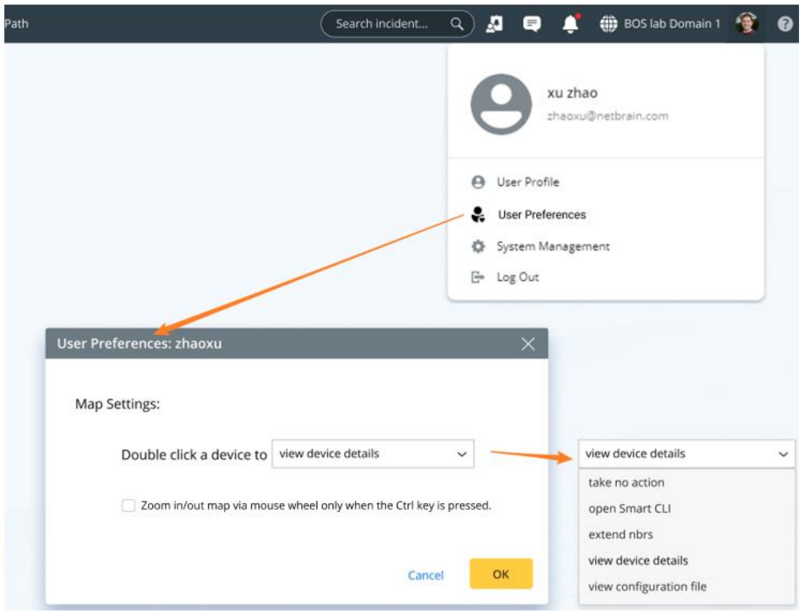
- In the **Map Page setting**, Add **Apply to Other Maps** so that users can bulk apply the settings to other maps.



- Users can set the background color. White is the default color.



- Users can set up **User Preference** to define the behavior of double-clicking a device and whether scrolling the mouse wheel to zoom in/out a map requires pressing the **Ctrl** key.



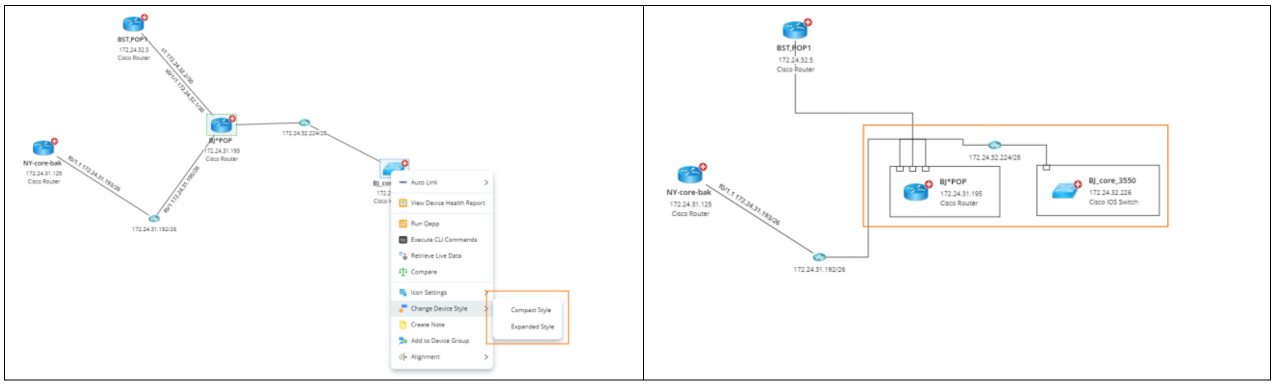
User Preferences: zhaoxu

Map Settings:

Double click a device to: view device details

☐ Zoom in/out map via mouse wheel only when the Ctrl key is pressed.

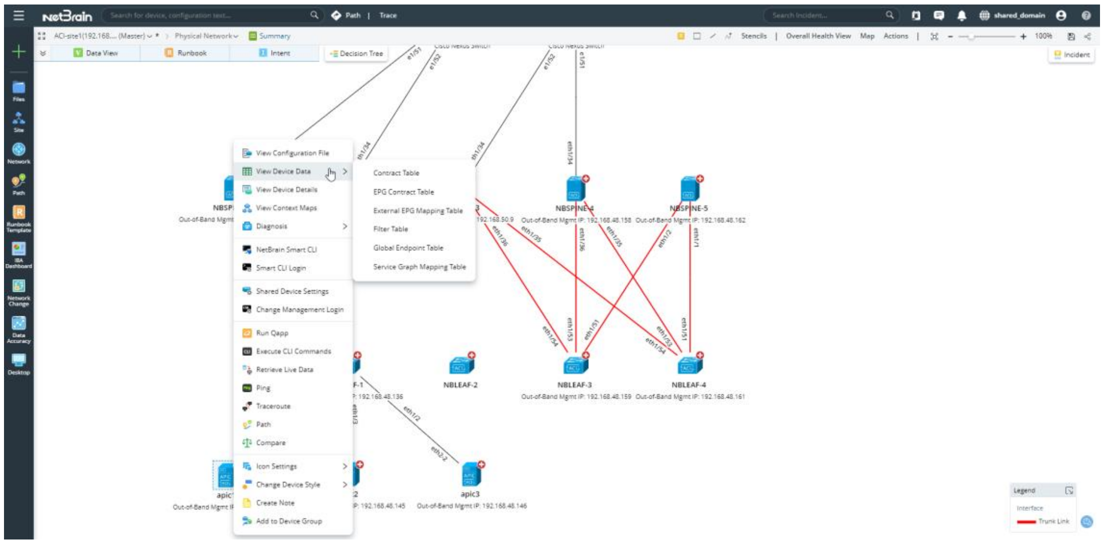
- Allow users to change the device style (compact or expanded) of the selected device(s) instead of all devices on the map.



- Users can decide whether to release editing rights when closing the map. This option will take effect at map level and applies to shared map/site map/device group map.

6.8.2 ACI Leaf Global NCT Moved To APIC

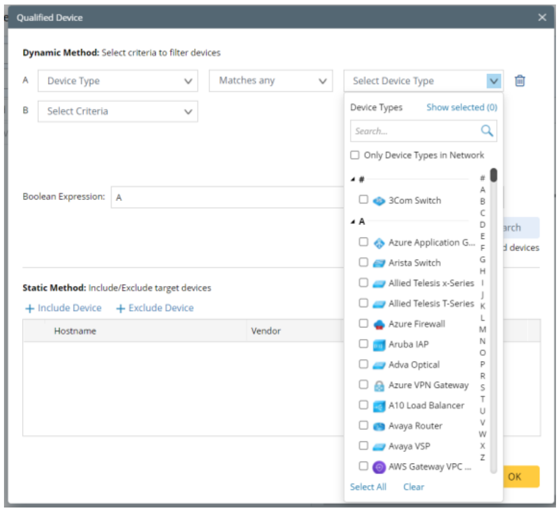
In.1, the ACI leaf global NCT will be moved to the APIC server, reducing redundant data storage. The user can view the Global Data Tables in Cisco ACI. Users can also view via View Device Data> Related NCTs in spine and leaf.



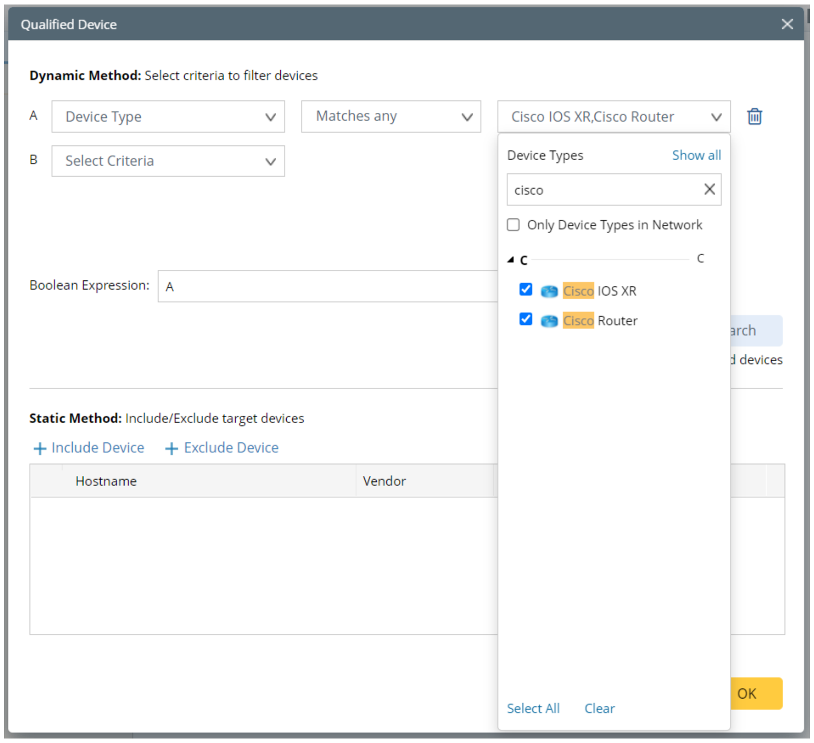
6.8.3 Select Device Type Improvements

The system now supports more than 200 device types, and sometimes it is not easy to locate a device type. Therefore, V10.1 makes the following improvements:

- all supported device types are sorted in numerical and alphabetical orders and add an option **Only Device Types in Network** so that only device types in the current domain will be displayed.

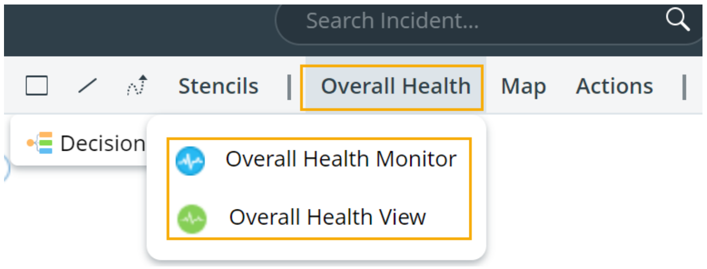


- Support searching device type with the keyword.
- Support switching between **Show selected** and **Show all**. While selecting the device type, the user can click **Show selected** to see what device types have been selected; afterward, the user can click **Show all** to continue the selection.



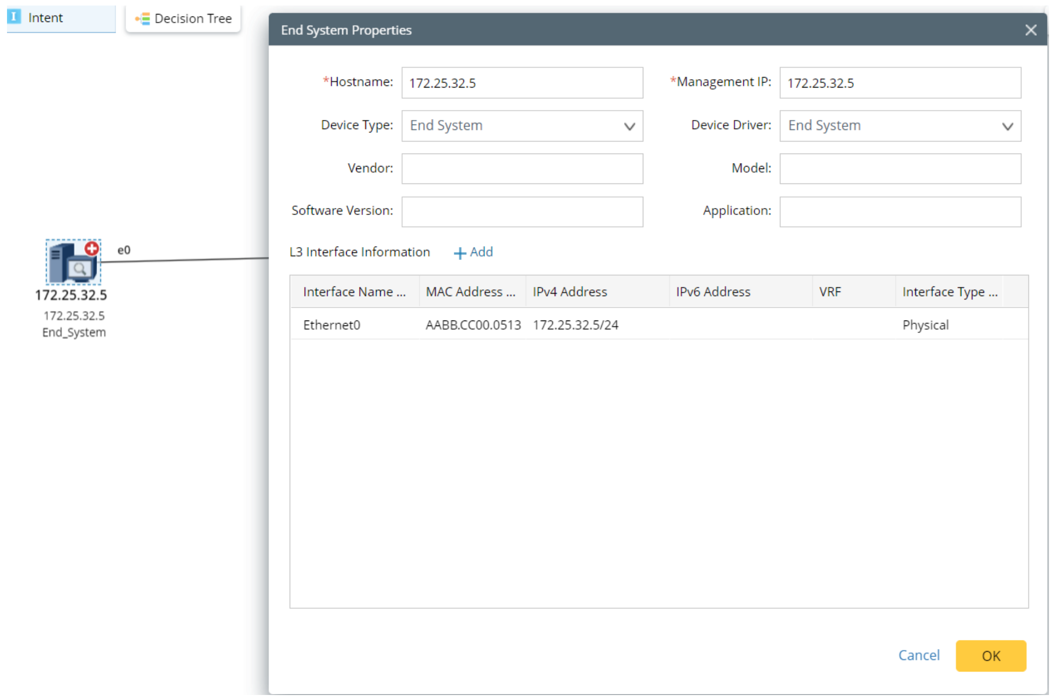
6.8.4 Overall Health Monitoring Adjustments

To avoid users' confusion, **Overall Health Monitoring** will be equal to **Overall Health View** (DVT), not including **Overall Health Monitor** (Qapp). Users can still use **Overall Health Monitor** Qapp as a general built-in Qapp.



6.8.5 Unknown End System Improvements

V10.1 allows users to resolve the DNS name or add an unknown IP as an end system directly from the map by setting its property.



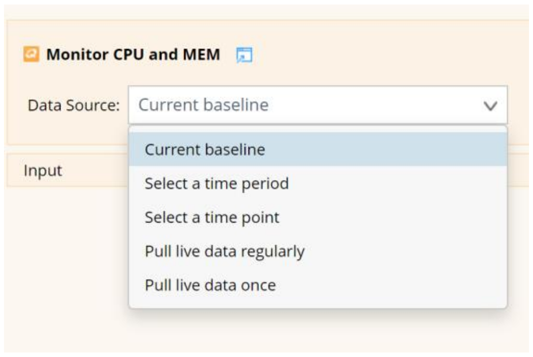
*Hostname:	172.25.32.5	*Management IP:	172.25.32.5
Device Type:	End System	Device Driver:	End System
Vendor:		Model:	
Software Version:		Application:	

Interface Name ...	MAC Address ...	IPv4 Address	IPv6 Address	VRF	Interface Type ...
Ethernet0	AABB.CC00.0513	172.25.32.5/24			Physical

6.8.6 Qapp Improvements

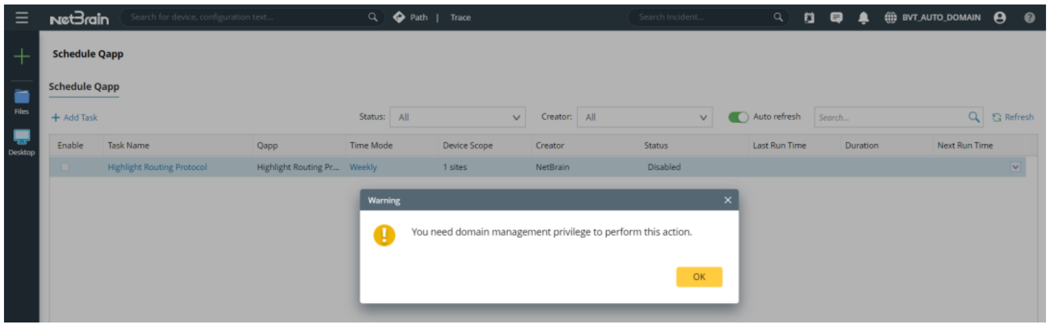
V10.1 made the following improvements on Qapp:

- When defining Qapp, users can specify which data source(s) to apply and then set a specific data source as the default. So, the Qapp won't fail to run due to wrong data source selection, or users can directly run Qapp with its default data source.


Monitor CPU and MEM

Data Source: Current baseline

- End users can view Qapp task results from the **Schedule Qapp** page. He can select **Run Now**, view results, and **Export to CSV**. However, users must have domain management privilege to edit, disable/ enable, delete tasks created by other users. And they cannot stop tasks currently run by other users.



Status: All

Creator: All

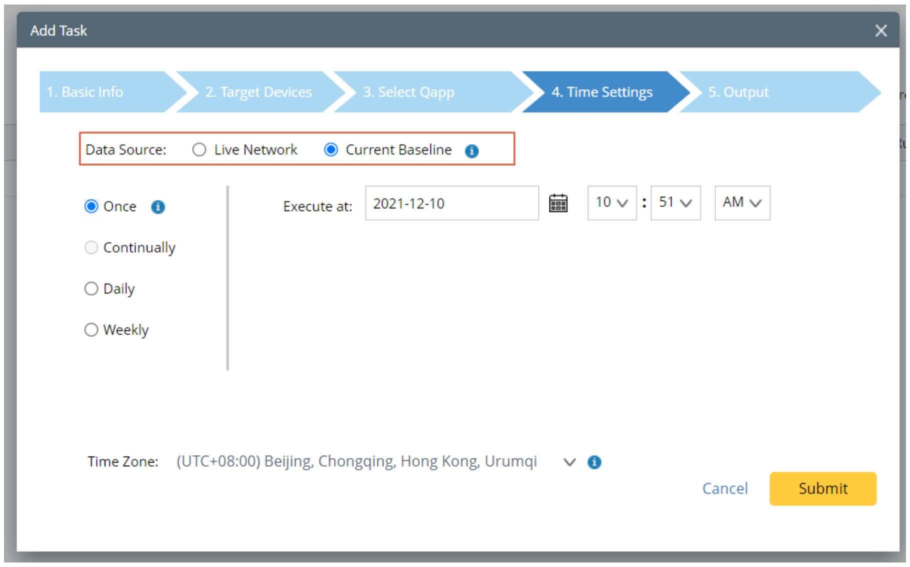
Enable	Task Name	Qapp	Time Mode	Device Scope	Creator	Status	Last Run Time	Duration	Next Run Time
	Highlight Routing Protocol	Highlight Routing Pr...	Weekly	1 sites	NetBrain	Disabled			

Warning

You need domain management privilege to perform this action.

OK

- Users can select the current baseline as the data source to run the scheduled Qapp for compliance check and report.



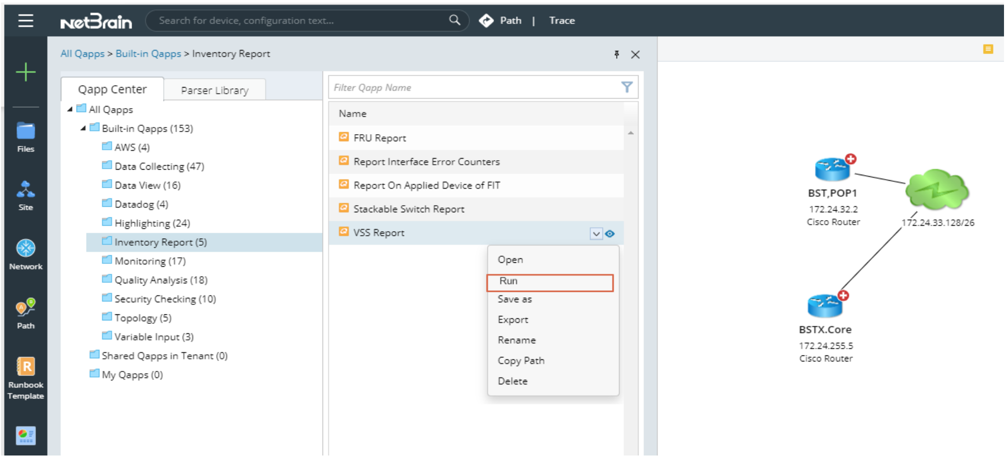
Data Source: Live Network [✓ Current Baseline]

[✓ Once] Continually Daily Weekly

Execute at: 2021-12-10 10 : 51 AM

Time Zone: (UTC+08:00) Beijing, Chongqing, Hong Kong, Urumqi

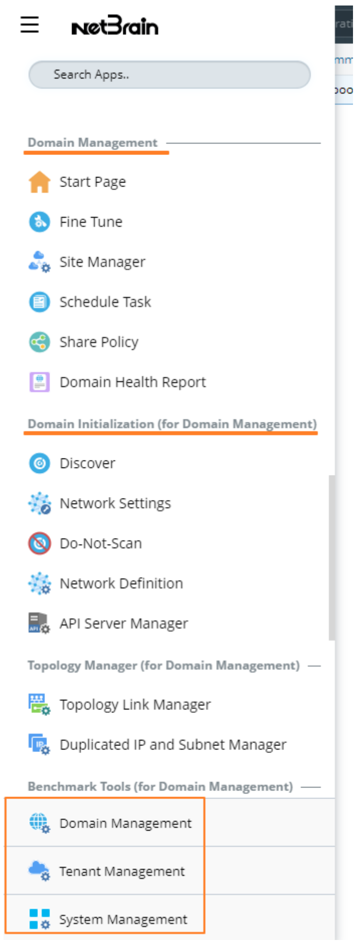
- Allow users to run the Qapp directly from the Qapp Center. The Qapp will be added to the runbook and run.



6.8.7 Admin UI Improvements

In the previous version, the domain admin, tenant, and system admin pages are hard to navigate. Especially, the operation drop-down menus have too many levels so that users cannot find the things. V10.1 made the following improvements:

- Add frequently used domain management functions in the start menu of the main UI and provide three links at the bottom of the start menu for the domain management page, current tenant management page, and system management page if the user has the corresponding privileges.



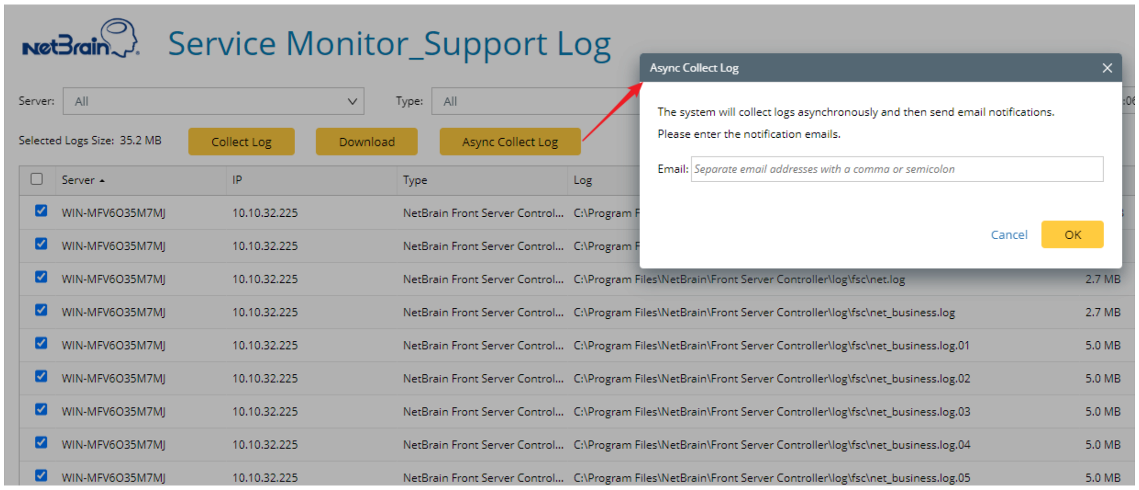
- The domina management start menu has all the functions in domain management and three links to Desktop (main UI), current tenant management page, and system management page.
- The system management start menu has all functions in system management and a link to any tenant management.
- The tenant management start menu has all functions in tenant management and includes a link to the system management.

6.8.8 Supported Operations of Device Type Adjustments

Some of the supported operations configured on the device type are no longer meaningful and removed from the definition. A new operator, Data Accuracy Wizard (DAW), is added. For example, the end system does not need to support the DAW function.

6.8.9 Service Monitor Improvements

The log collection of the service monitor may take a long time if the log is large. IEv10.1 adds an Async Collect Log button to allow users to enter an email address and be notified without waiting for the process to finish.



Selected Logs Size: 35.2 MB

The system will collect logs asynchronously and then send email notifications.

Please enter the notification emails.

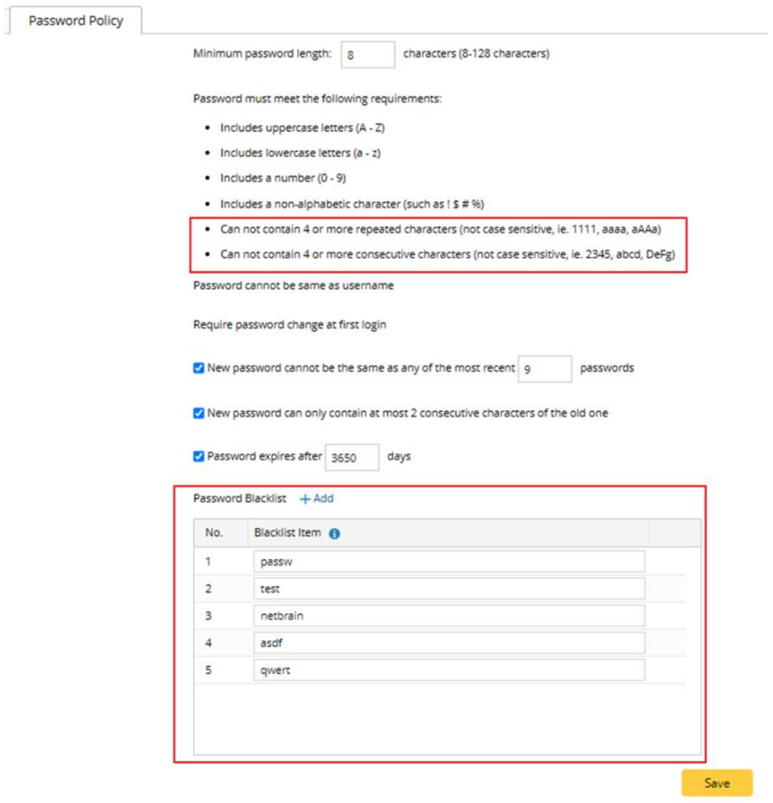
Email: *Separate email addresses with a comma or semicolon*

	Server	IP	Type	Log	
✓	WIN-MFV6O35M7MJ	10.10.32.225	NetBrain Front Server Control...	C:\Program F	
✓	WIN-MFV6O35M7MJ	10.10.32.225	NetBrain Front Server Control...	C:\Program F	
✓	WIN-MFV6O35M7MJ	10.10.32.225	NetBrain Front Server Control...	C:\Program Files\NetBrain\Front Server Controller\log\fsc\net.log	2.7 MB
✓	WIN-MFV6O35M7MJ	10.10.32.225	NetBrain Front Server Control...	C:\Program Files\NetBrain\Front Server Controller\log\fsc\net_business.log	2.7 MB
✓	WIN-MFV6O35M7MJ	10.10.32.225	NetBrain Front Server Control...	C:\Program Files\NetBrain\Front Server Controller\log\fsc\net_business.log.01	5.0 MB
✓	WIN-MFV6O35M7MJ	10.10.32.225	NetBrain Front Server Control...	C:\Program Files\NetBrain\Front Server Controller\log\fsc\net_business.log.02	5.0 MB
✓	WIN-MFV6O35M7MJ	10.10.32.225	NetBrain Front Server Control...	C:\Program Files\NetBrain\Front Server Controller\log\fsc\net_business.log.03	5.0 MB
✓	WIN-MFV6O35M7MJ	10.10.32.225	NetBrain Front Server Control...	C:\Program Files\NetBrain\Front Server Controller\log\fsc\net_business.log.04	5.0 MB
✓	WIN-MFV6O35M7MJ	10.10.32.225	NetBrain Front Server Control...	C:\Program Files\NetBrain\Front Server Controller\log\fsc\net_business.log.05	5.0 MB

When the system finishes collecting the log, a notification will be sent to this email address. The users can click the link in the email to download logs.

6.8.10 Security Related Enhancements

- Added the support for Alma and Rocky Linux. Updates to CENTOS are no longer supported, and so for kernel versions 8.5 and above, support for Alma and Rocky Linux was added for new OS updates (including security issues/vulnerabilities).
- Enhancements to automatic update/apply capabilities: security considerations included for IE update management.
  - Improved password policy: new configurable password blacklist. Add two hardcode rules that the password must meet.



Password Policy

Minimum password length: 8 characters (8-128 characters)

Password must meet the following requirements:

- Includes uppercase letters (A - Z)
- Includes lowercase letters (a - z)
- Includes a number (0 - 9)
- Includes a non-alphabetic character (such as ! $ # %)
- Can not contain 4 or more repeated characters (not case sensitive, ie. 1111, aaaa, aAAa)
- Can not contain 4 or more consecutive characters (not case sensitive, ie. 2345, abcd, DeFg)

Password cannot be same as username

Require password change at first login

✓ New password cannot be the same as any of the most recent 9 passwords

✓ New password can only contain at most 2 consecutive characters of the old one

✓ Password expires after 3650 days

Password Blacklist + Add

No.	Blacklist Item
1	passw
2	test
3	netbrain
4	asdf
5	qwert

Save

- Updated the following 3rd party library:

Library Name	**Previous version**	**New Version**
Log4j2	2.11.1	2.17.1
JDK	11.0.9+11	11.0.14.1+1
Uri.js	1.19.6	1.19.8
Elasticsearch (C / S)	6.8.6 / 6.8.12	6.8.9 / 6.8.23
Urllib3	1.26.4	1.26.8
Redis	6.0.13	6.2.6
Follow redirects	1.5.10	1.14.7
Axios	0.19.2	0.25.0
HighCharts	8.2.2	9.2.2
MongoDB	4.0.20	4.0.28
RabbitMQ	3.8.16	3.8.19
Underscore	1.9.1	1.12.1
OpenSSL	1.1.1j	1.1.1l

7 Appendix

7.1 Version Compatibility and Upgrade

The following IE versions can be upgraded to v10.1:

- V10.0 and 10.0a
- V8.0, 8.01, 8.02, and 8.03
- V7.1, 7.1a, 7.1a1, 7.1a2 and 7.1a3
- V7.0b and 7.0b1

Due to the license adjustments, the license must be reactivated after upgrading to 10.1. Also, the benchmark data of v7.x cannot be used, and users must run a full benchmark after upgrading.

7.2 Performance Improvement

The test results on NetBrain lab show improvements of benchmark tasks compared with the prior version:

Test Scenario	**V10.0a**	**V10.1**
Benchmark 45,000 simulated devices	5 hours and 3 mins	2 hours and 54 mins
Build L2 topology of 45,000 simulated devices	1 hour and 2mins	31 mins
Build L3 topology of 45,000 simulated devices	31 mins	2 mins

Also, we compare the performance of Qapp and NI to implement the same function. The results show that NI is six times faster than Qapp.