NetBrain® Integrated Edition

8.02 Release Notes

Version 8.02 | Last Updated 2020-02-13

Contents

1. Summary

NetBrain Integrated Edition 8.02 (IEv8.02) is a minor release that introduces many feature enhancements and bug fixes.

- Enhanced Reference Flow and Functionalities:
  - Event-driven Automation Framework to Proactively Integrate with 3rd-party Systems.
  - Enhanced Search Capability to Search for 5 New Objects: Network Context, Path, Event, One-IP Table, ACI Endpoint Table.
  - Calculate a Multicast Path from a Receiver to a Source
  - Document Network Change and Runbook Automation Word
  - 5 Enhancements to Reference Flow, such as New Hints for Golden Baseline Alerts, Messaging Golden Baseline Alerts, etc.
  - New Domain Health Report
  - 6 Map Enhancements, such as Mapping One-IP Table Entry with Drag-and-Drop, Narrowing Down Auto-link Scope and Count, Custom Node Icons, Free Test for Network Design/Troubleshooting Annotation, etc.
  - 4 Site Enhancements, such as Excluding Specified Device Types from Site Assignment, Allowing Hiding Neighbor Sites from a Site Map, etc.
  - 4 Enhancements to Domain Maintenance
  - 3 Enhancements to Service Monitor, such as Scheduling Auto-Restart for Front Server (Controller), Alerting for Stopped Services, Log Analysis for System Health, etc.
  - 2 Enhancements to Change Management, such as Scheduling a Network Change, Using Variables to Protect Data Security.
  - 3 More Feature Enhancements
- Platform Enhancements
  - 3 Enhancements to Platform Framework, such as Multi-CLI Mode support, Configuration Retrieval for SDN Nodes via TechSpec, New APIs for Plugin/Qapp.
  - 3 Extensions Based on Enhanced Platform Framework, such as QoS Path Support, Cisco Wireless L2 Path Support, Hostname Change Support to VDC Devices.
  - 3 Enhancements to Path Framework, such as Enhanced Path Debuggability, etc.
  - Fully Extensible MPLS Cloud Framework
  - 2 Enhancements to L2 Topology, such as Enhanced L2 Topology for SDN, etc.
  - 8 New REST APIs
- 9 Security Enhancements and Performance Enhancements

2. New Features and Feature Enhancements

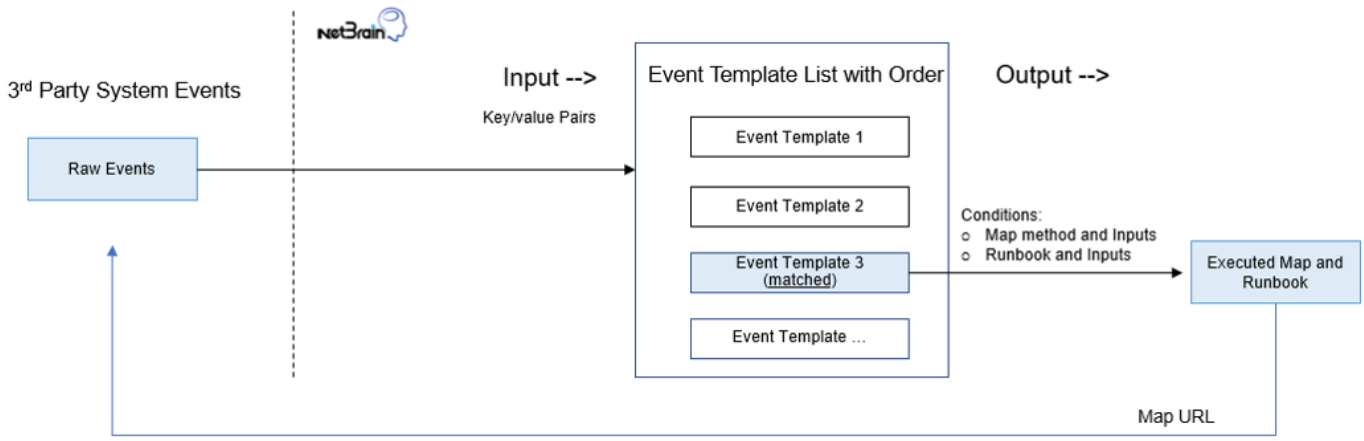
2.1. New Event-driven Automation Framework

Event-driven Automation is a new feature in v8.02, integrating with 3rd-party systems by processing triggered events, mapping problem areas, and executing runbooks based on predefined event conditions. With this capability, users only need to define general event templates in the NetBrain system, and they will process event analysis when receiving the 3rd-party event.

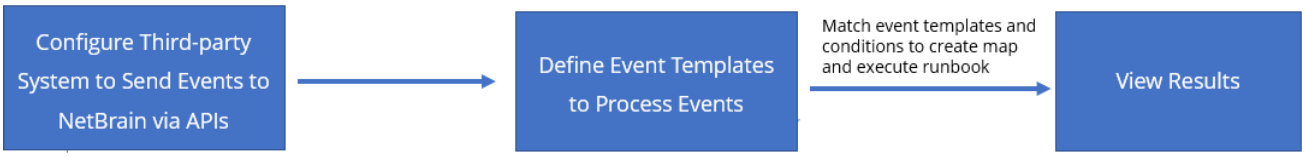
Event-driven Automation brings benefits as follows:

- It eases the burden for power users and 3rd-party system admin. They do not have to write lots of scripts in a 3rd-party system to call NetBrain maps or runbooks event by event. Instead, they only need to enable the 3rd-party system to send events to their NetBrain system directly and track event status by using REST APIs.
- It only has a minimal requirement of script supportability for a 3rd-party system. A system that can call REST APIs to send event data to NetBrain is ready to use NetBrain's automation capabilities.
- It improves the NetBrain system capability to process a large number of events by transferring task execution from Web API Server to Worker Server. It also enables horizontal scalability to improve event processing capacity.

Figure: Event-driven Automation Data Flow



Use Flow



2.1.1.Configure Third-party System to Send Events to NetBrain via APIs

The system provides two REST APIs to send events from a 3rd-party system to NetBrain and track its status.

- Drive Events – used to send all events generated in a 3rd-party system to a NetBrain domain.
- Track Event Status – used to get the execution status of an event.

Example: Define business rule in ServiceNow to send events to NetBrain IE.

```
import requests
import json
import time
import requests.packages.urllib3 as urllib3
urllib3.disable_warnings()
# Need to install requests package for python
# pip install requests

user = "admin"                                  # account to log in to your NetBrain Domain
pwd = "admin"                                  # password
host_url = "http://10.10.0.29"                  # The URL of your NetBrain Domain
headers = {'Content-Type': 'application/json', 'Accept': 'application/json'}
headers1 = {'Content-Type': 'application/json', 'Accept': 'application/json'}
'''
Get token for netbrain
'''
TENANT = 'Initial_Tenant'
DOMAIN = 'domain1'
def getTokens(user,password):
    login_api_url = r"/ServicesAPI/API/V1/Session"
    Login_url = host_url + login_api_url
    data = {
        "username": user,
        "password": password
    }
    token = requests.post(Login_url, data=json.dumps(
        data), headers=headers, verify=False)
    if token.status_code == 200:
        print(token.json())
        return token.json()["token"]
    else:
        return "error"
# get token
token = getTokens(user,pwd)
headers["Token"] = token
```

```
def get_tenant_domain_id():

    tenant_id_url = '/ServicesAPI/API/V1/CMDB/Tenants'
    full_url = host_url + tenant_id_url
    data = requests.get(full_url,headers=headers,verify=False)
    # tenant_id = '78a825ef-24bd-729d-f56f-a1ad2b79f2ff'
    # domain_id = '36700aff-c585-4f23-95eb-8ea00214b778'
    print(data.json())
    if data.status_code == 200:
        for tenant in data.json()['tenants']:
            if TENANT == tenant['tenantName']:
                tenant_id = tenant['tenantId']
        if tenant_id:
            domain_id_url = '/ServicesAPI/API/V1/CMDB/Domains'
            full_domain_url = host_url +domain_id_url
            domain_data =
requests.get(full_domain_url,params={'tenantId':tenant_id},headers=headers,verify=False)
            print(domain_data.json())
            if domain_data.status_code == 200:
                for domain in domain_data.json()['domains']:
                    if DOMAIN == domain['domainName']:
                        domain_id = domain['domainId']
        return tenant_id,domain_id
    else:
        return tenant_id,domain_id

tenant_id,domain_id = get_tenant_domain_id()
print(tenant_id,domain_id)
headers["TenantGuid"]= tenant_id
headers["DomainGuid"]= domain_id

def Logout():
    logout_url = "/ServicesAPI/API/V1/Session"
    time.sleep(2)
    full_url = host_url + logout_url
    body = {
        "token": token
        }
    result = requests.delete(full_url, data=json.dumps(body), headers=headers, verify=False)
    print('Logout: ' + str(result.json()))
    if result.status_code == 200:
        print("LogOut success...")
    else:
        data = "errorCode" + "LogOut API test failed... "
        return result.json()
# Trigger API function

def PublishEvent(Event_Data):
    # Trigger  API url
    API_URL = r"/ServicesAPI/API/V1/CMDB/EventDriven/Events"
    # Trigger API payload
    print(headers)
    api_full_url = host_url + API_URL
    print('api_full_url: ' + api_full_url)
    api_result = requests.post(api_full_url, data=json.dumps(Event_Data), headers=headers,
verify=False)
    if api_result.status_code == 200:
```

```
        return api_result.json()
    else:
        return api_result.json()


if __name__ =="__main__":
    #tenant_id,domain_id = get_tenant_domain_id()
    #print(tenant_id,domain_id)
    # tenant_id = '0b7eb490-d9cf-aacc-672c-ff9d58a47032'
    # domain_id = '53e4b108-086e-4b6f-95b8-ee23bd7d142a'
    Event_Data = {
        "parent": "",
        "u_path_analysis_set": "",
        "made_sla": "true",
        "cause_by": "",
        "watch_list": "",
        "u_nb_task": "",
        "upon_reject": "cancel",
        "sys_updated_on": "2019-06-27 15:54:14",
        "child_incidents": "0",
        "approval_history": "",
        "skills": "",
        "number": "INC0011879",
        "u_destination_port": "1234",
        "u_source_ip_new": "1.1.1.1",
        "resolved_by": "chris.zhao",
        "opened_by": {
            "link": "https://ven01749.service-now.com/api/now/table/sys_user/22121da321adf",
            "value": "232s9i2asko92asdf232322d098s"
        },
        "user_input": "",
        "state": "2",
        "knowledge": "false",
        "active": "true"
    }
print(PublishEvent(Event_Data))
```

2.1.2.Define Event Templates to Process Events

An event template is a custom executor in the NetBrain system to process events from a specified 3rd-party system. After NetBrain receives an event, it will check the qualification defined in each event template and use the qualified event template to decide whether to map and execute a runbook for the event based on conditions.

An event template is defined through python scripts and contains the following functions:

- **Define Qualification** – define the criteria that an event template can be applied to an event. There are two methods to define a qualification:
  - **Regular expression**

```
"type”:.*?servicenow.*?
```

- **Def qualify**

```
def qualify(raw_event):
    raw_event_json = nbjson.loads(raw_event)
    if "number" in raw_event_json and "INC" in
raw_event_json["number"]:
        return True
    else:
        return False
```

- **Define Condition and Automation (def translate)** – used to translate an event to a NetBrain task and define conditions and corresponding drill-down actions. The actions include:
  - Parse the values of variables required for map and runbook automation.

```
map_setting = mapSettingHelper.build_device_map_setting("device_name", True,
"L3_Topo_Type")
```

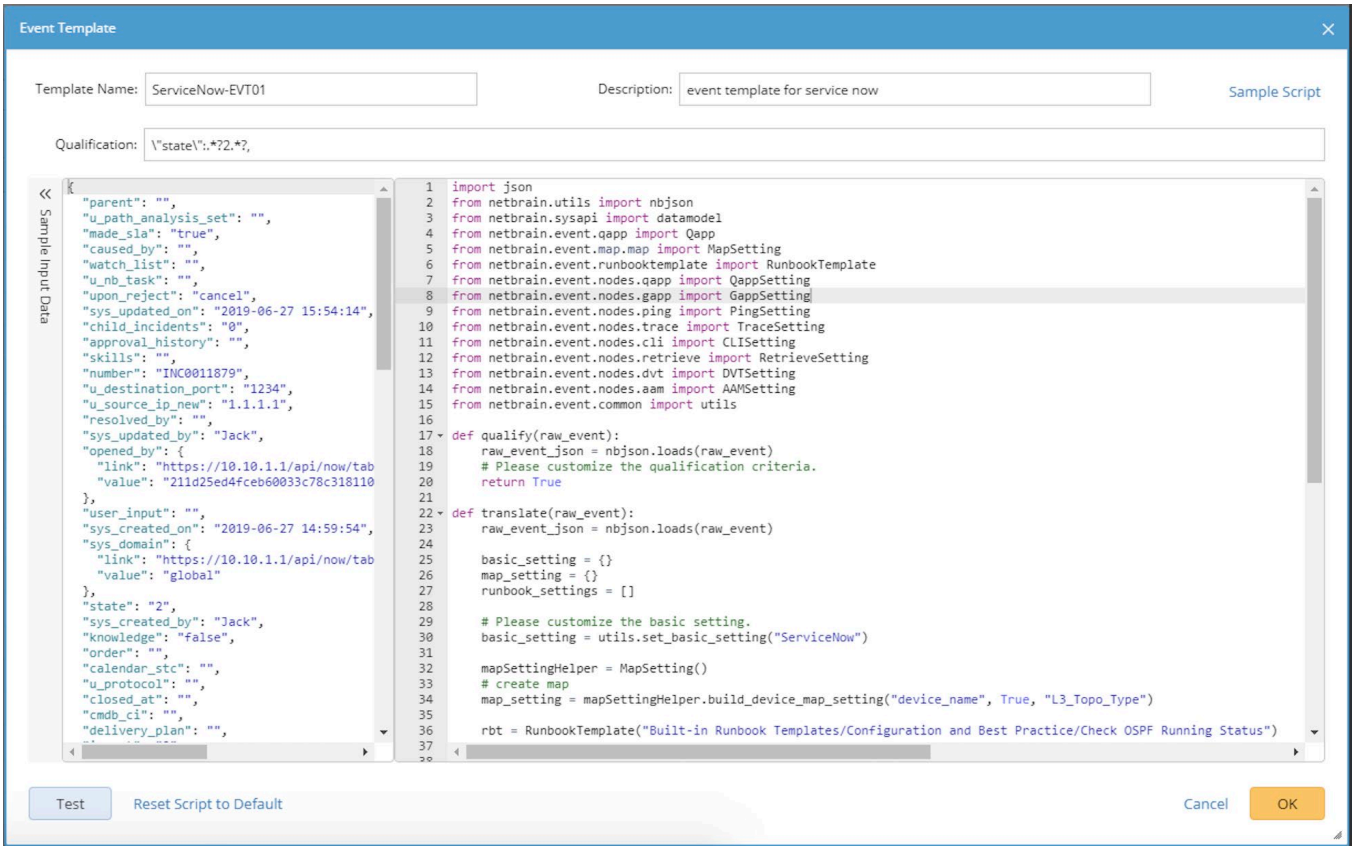
  - Define the conditions for automation.

```
if True:
    mapSettingHelper = MapSetting()
    map_setting = mapSettingHelper.build_site_map_setting("device_name", False)
    Pass
```

  - Define the input values required to draw a map or execute a runbook, and variable mapping.

```
# node name: 2. Retrieving the CLI commands of Failover, type: Execute CLI Commands
Node
rb_node = rbt.get_rbt_node("632653e6-4676-4f94-927f-86a6318f71e5")
cli_command_node = CLISetting(rb_node)
#cli_command_node.set_cli_commands(["show failover","show monitor-interface","show
failover interface"])
rbt.update(cli_command_node.value())
```

Example: Define an event template for events from ServiceNow.



Event Process Logic:

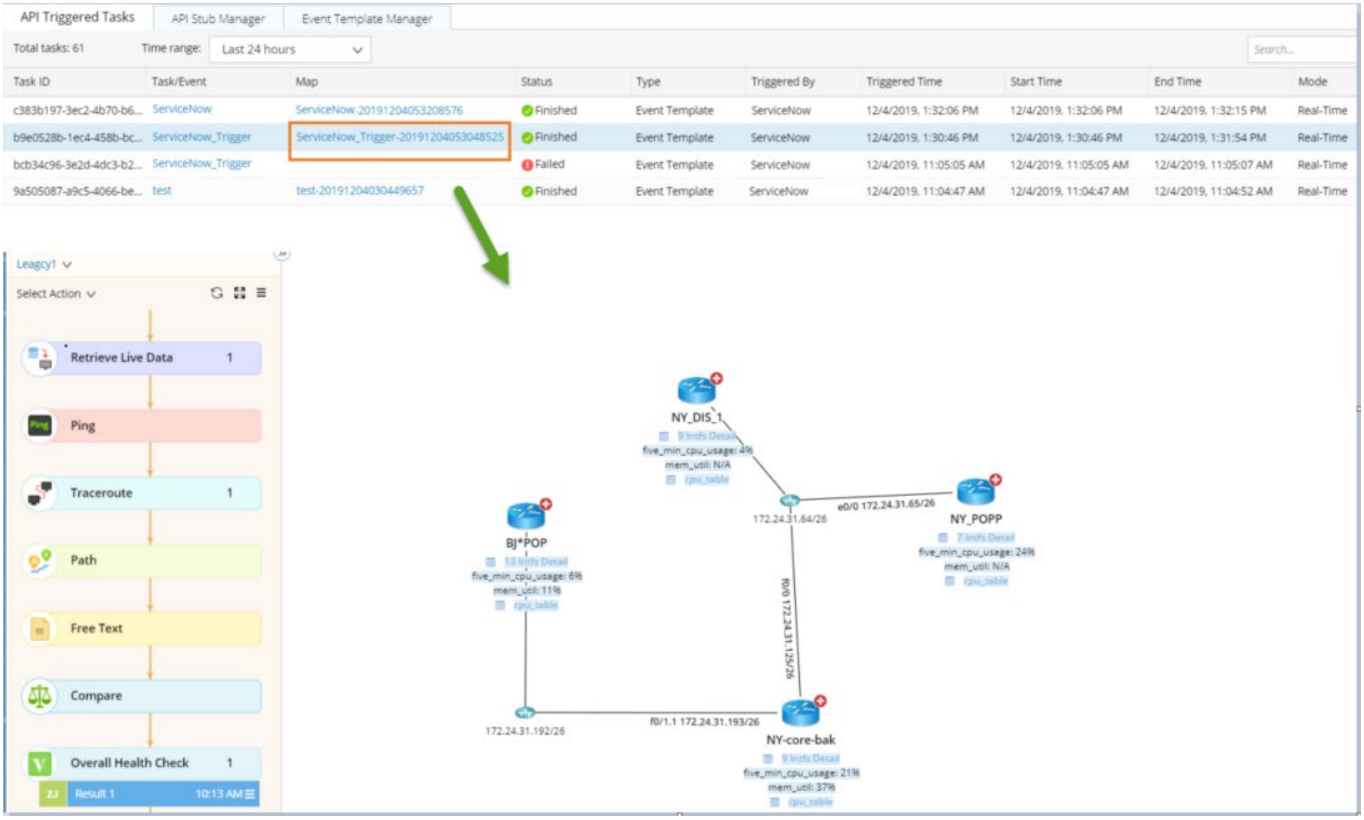
After receiving an event, the system will:

1. Check all event templates in order until finding the one matching the event conditions.
2. Execute data parsing, mapping and automation actions defined in the matched event template.
3. Return the result to the 3rd-party system in the form of a Map URL, and generates the corresponding task in NetBrain IE.

For more details on how to define an event template, refer to Online Help.

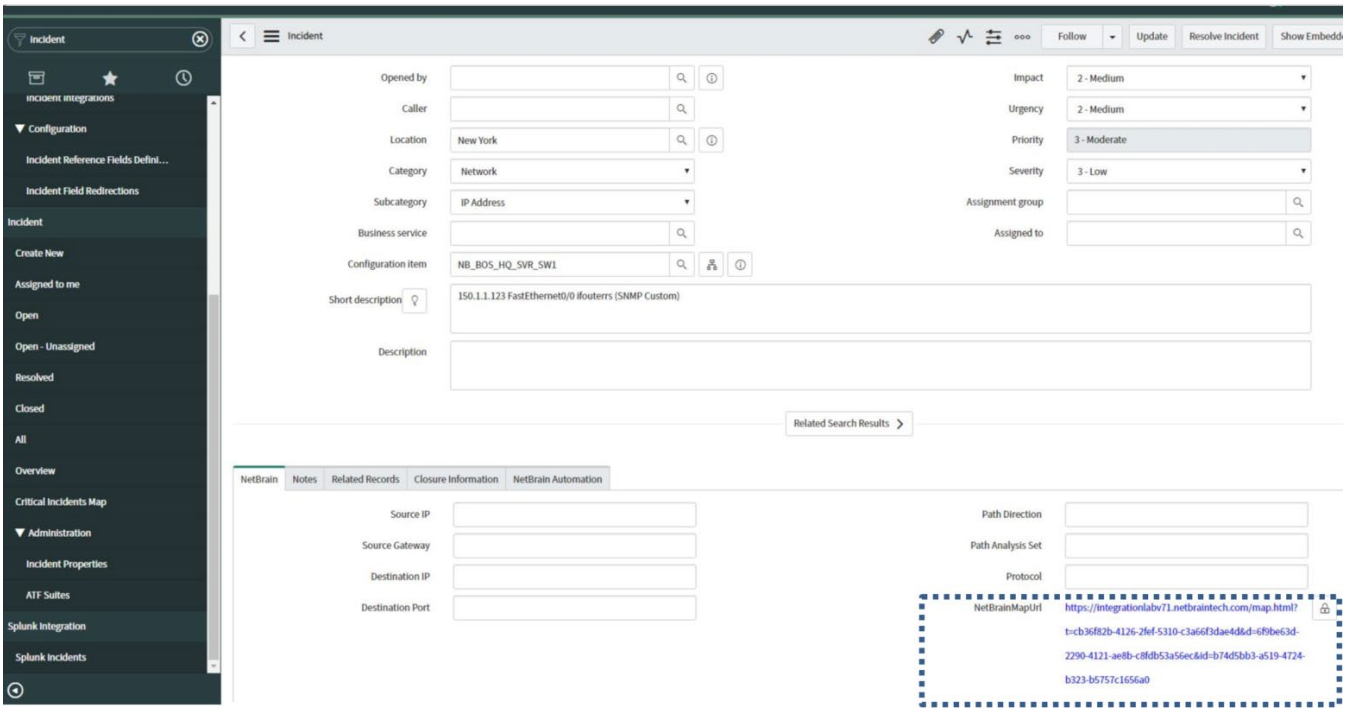
2.1.3.View Results in System Automation Manager

All 3rd-party events and results are recorded in the System Automation Manager of NetBrain IE. Once an event task is completed, end users can view the map as well as the data collected by the Runbook.



The map URL of an event will be sent back to the 3rd-party system as a response result.

Example: Map URL of ServiceNow incident:



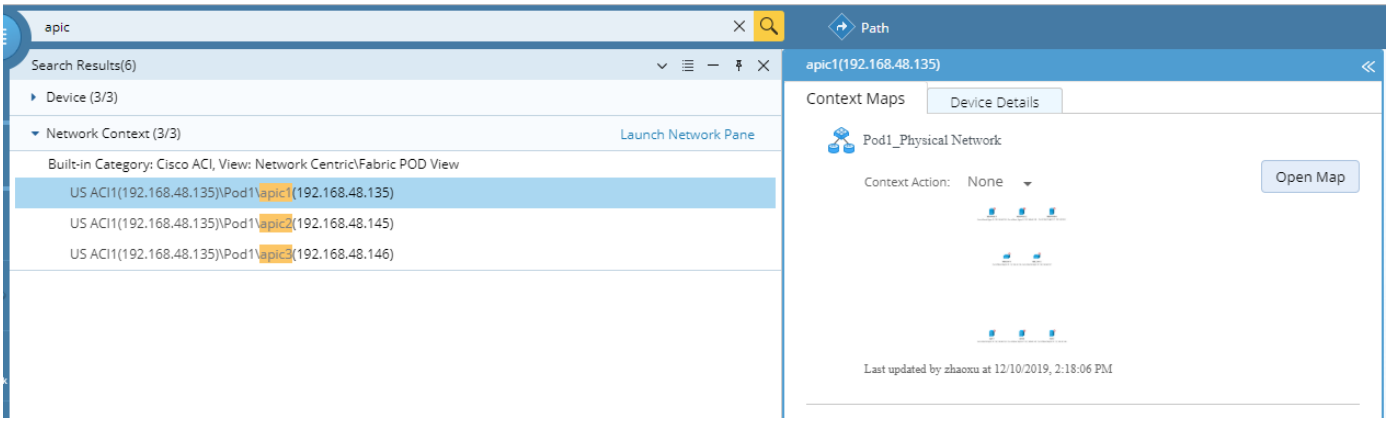
2.2. Search Engine Extension

IEv8.02 expands the scope for visual search and supports multiple search terms for each new category.

Search Object	Supported Search Terms
Network Context	▪ Network Tree Node Name
One-IP Table	▪ IP Address ▪ LAN Segment ▪ MAC Address ▪ Vendor ▪ Switch Port ▪ DNS Name ▪ Description
Path	▪ Hop IP/Hostname ▪ Source or Destination ▪ Application Name ▪ Path Name
Event	▪ Object (Device or Path) ▪ Event Message
Endpoint Table	▪ APIC Domain ▪ End Point ▪ IP address ▪ MAC Address ▪ Interface Name ▪ VLAN ▪ Learning Source ▪ EPG ▪ Reporting Controller ▪ Multicast Address

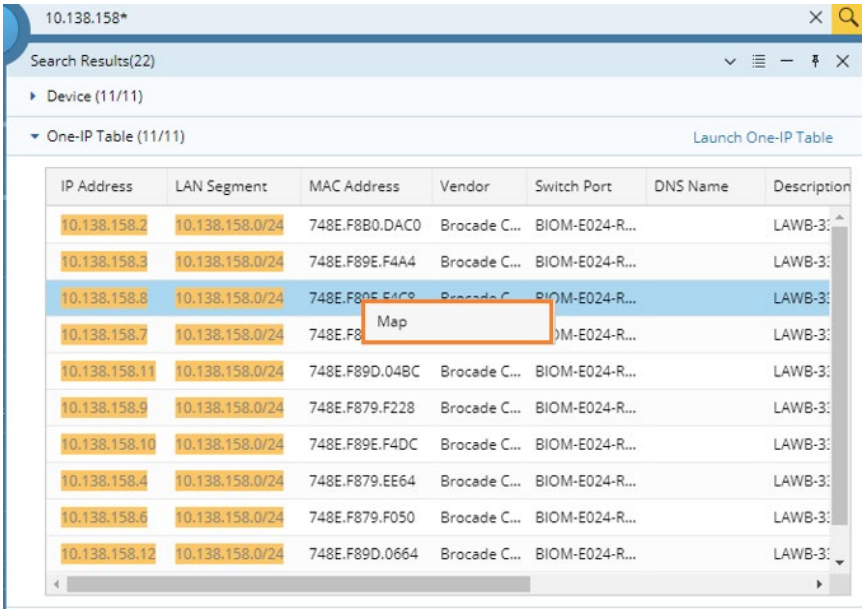
2.2.1.Search for Network Context Map

IEv8.02 extends the capability to show network contexts in the search results for an SDN node. With the context map, users can quickly understand the network design and perform drill-down actions.



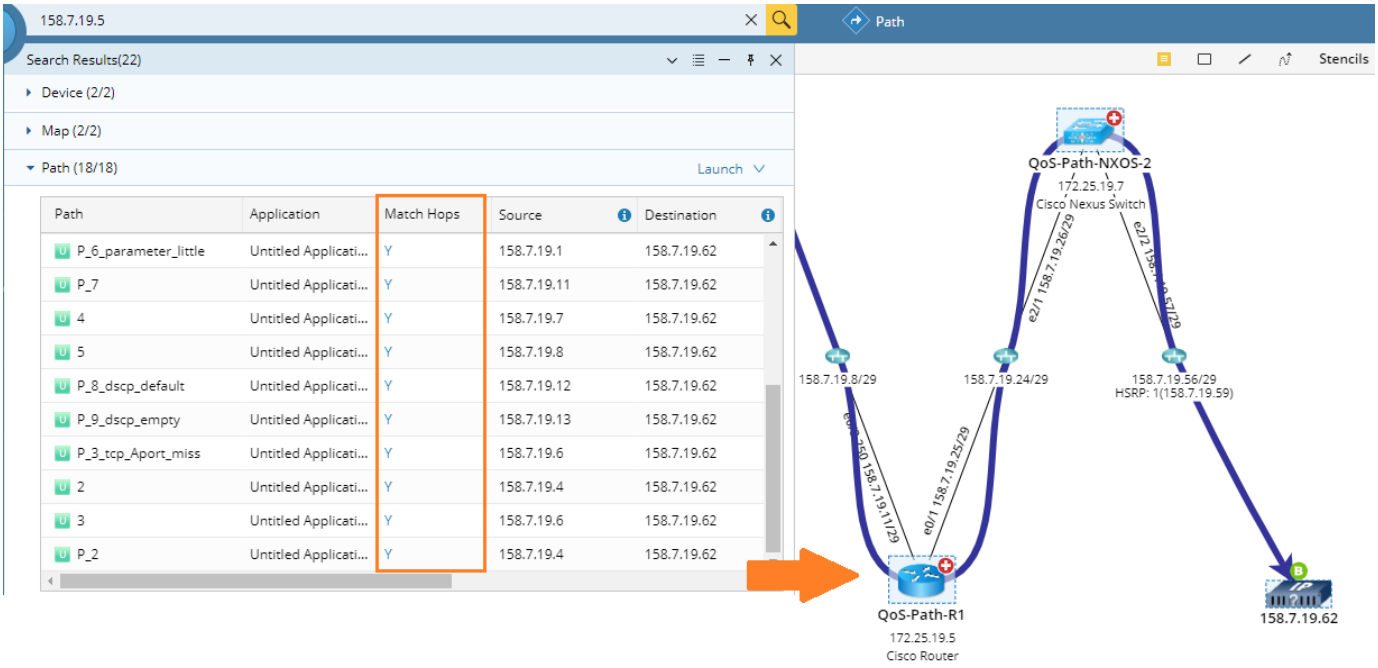
2.2.2.Search for One-IP Table

IEv8.02 adds the capability to show the One-IP table in the search results for an IP address. Users can directly view a device's L2 neighbors from the Switch Port column and map them out.



2.2.3.Search for A/B Paths across a Specific Hop

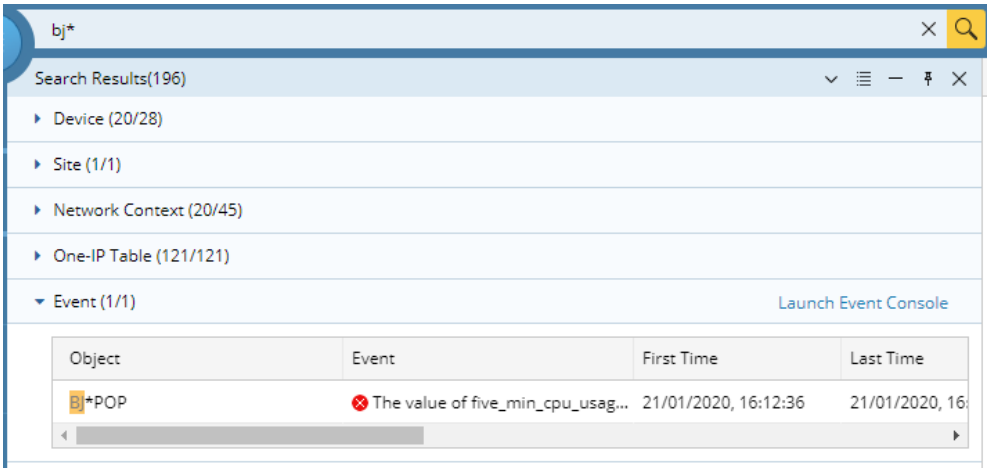
IEv8.02 adds the capability to search for all A/B paths crossing a specific hop by using the device name or IP address of the hop as the search term. For example, users can quickly address all the paths impacted by a problematic device, and map them out for drill-down troubleshooting.



Note: The search scope for A/B paths only includes those saved in the Path Browser.

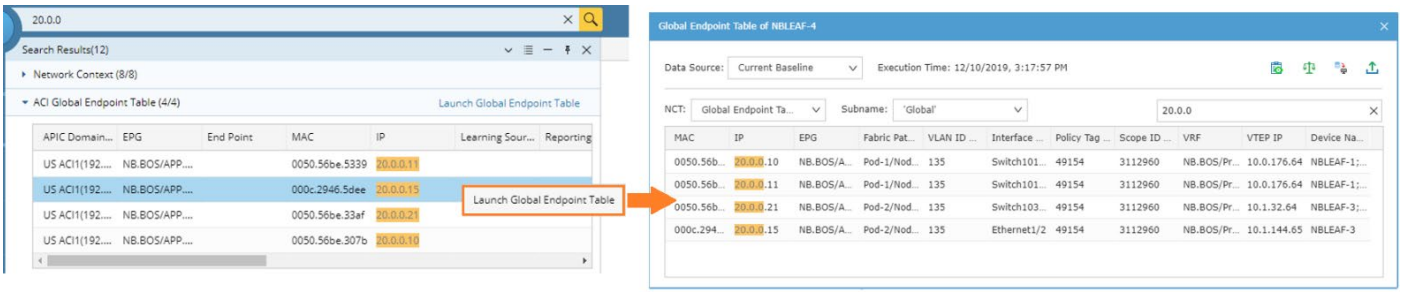
2.2.4.Search for Event

IEv8.02 adds the capability to search for events by using a keyword in the message as the search term. Users can start from search to map out the devices with alerts for drill-down troubleshooting.



2.2.5.Search for ACI Endpoint Table

Global Endpoint Table is mainly dedicated to path calculation for a Cisco ACI network, listing all endpoint information collected from an ACI fabric. IEv8.02 adds the capability to show the related endpoint table entry in the search results for an IP address. Users can quickly launch the full ACI Endpoint Table to drill down.



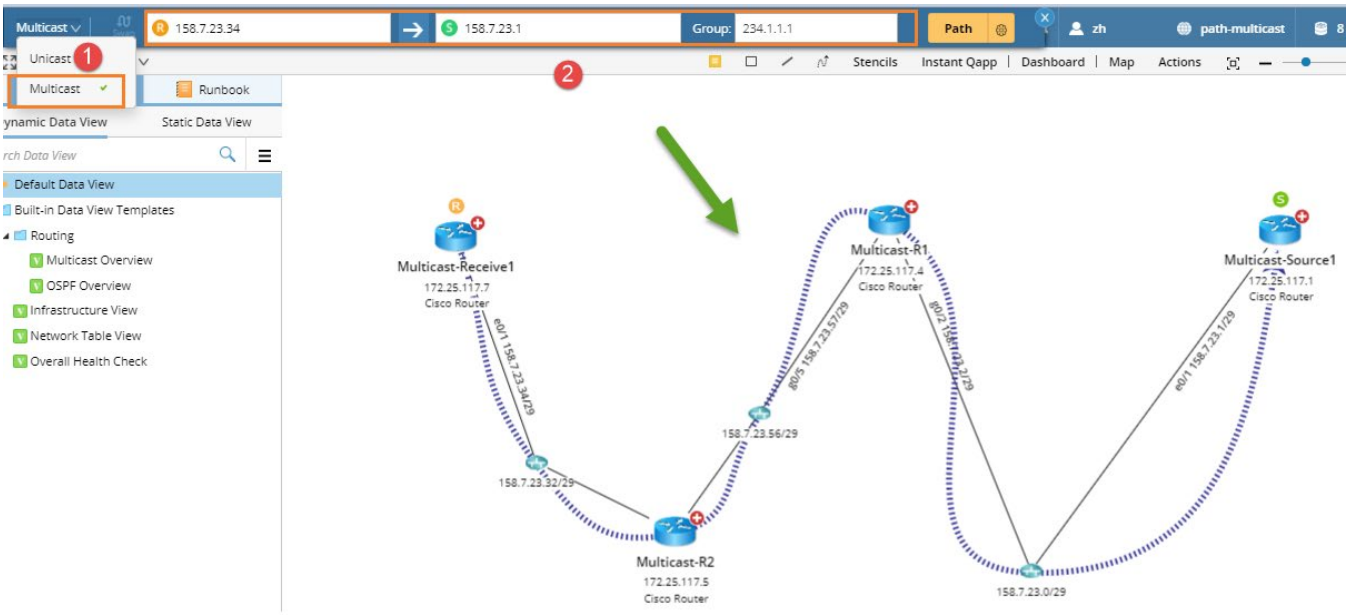
2.3. New Multicast Reverse Path Calculation

2.3.1.Calculate Multicast Reverse Path

When a problem occurs in a multicast network, users often need to troubleshoot the connection from a receiver to a source based on a specific group. In the previous versions, the system helped users troubleshoot multicast issues by drawing multicast paths via Qapps.

In IEv8.02, the system has extended the path framework, which enables users to calculate multicast paths. End users can specify a multicast group, source IP, and receiver IP to calculate a unidirectional multicast path from a receiver to a source.

A multicast reverse path includes four parameters: receiver, source, group, and gateway.



Multicast Path Calculation Logic

Compared with a unicast path, the following calculation logics are added for a multicast path.

- IEv8.02 adds an NCT “Multicast Route Table” and uses it to look up L3 next-hop devices during a path calculation.

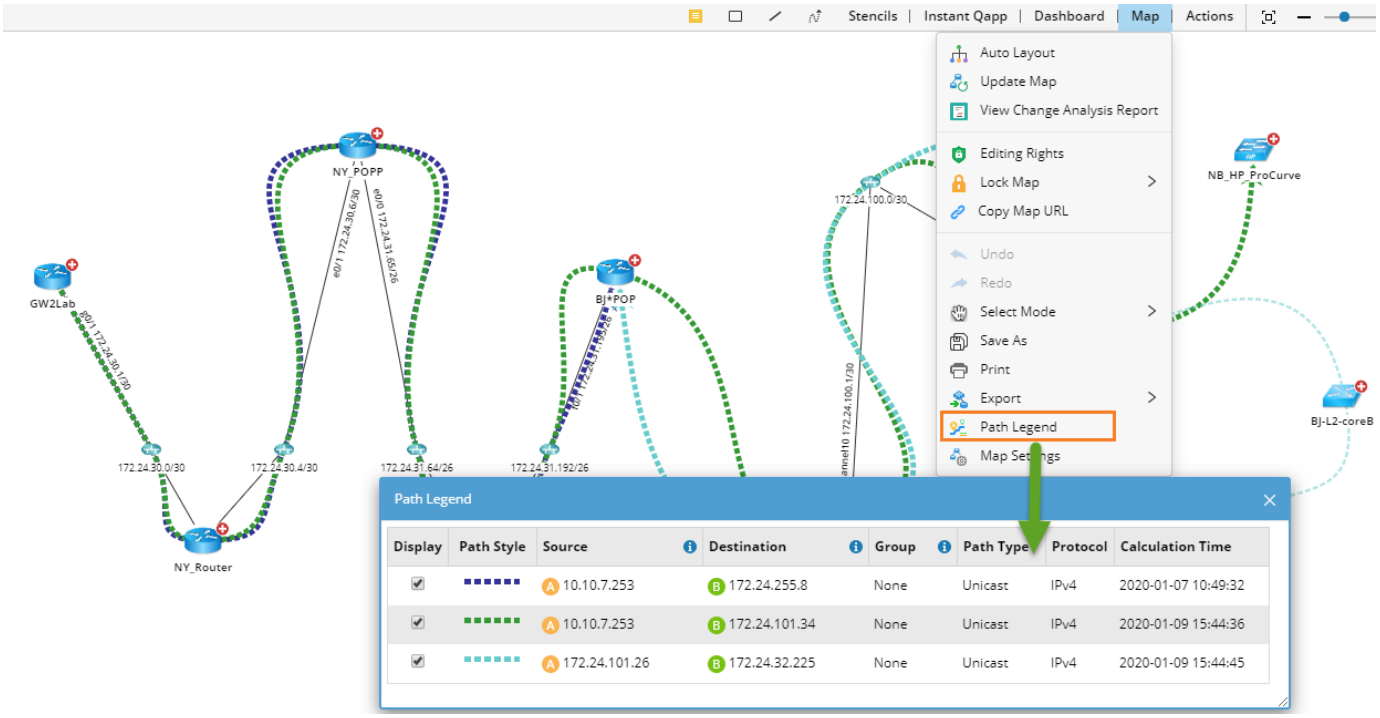
  Note: For the first-hop device, the system looks up its next-hop device based on the original routing table.

- When checking ACL/Policy on interfaces, the system checks whether the group IP as a destination is matched and continues path calculation based on the matching result.

2.3.2.View Path Legend on Map

IEv8.02 introduces the Path Legend function to differentiate paths on a map based on styles and types.

By default, the Path Legend pane is invisible. To view path legends, click **Map > Path Legend**.



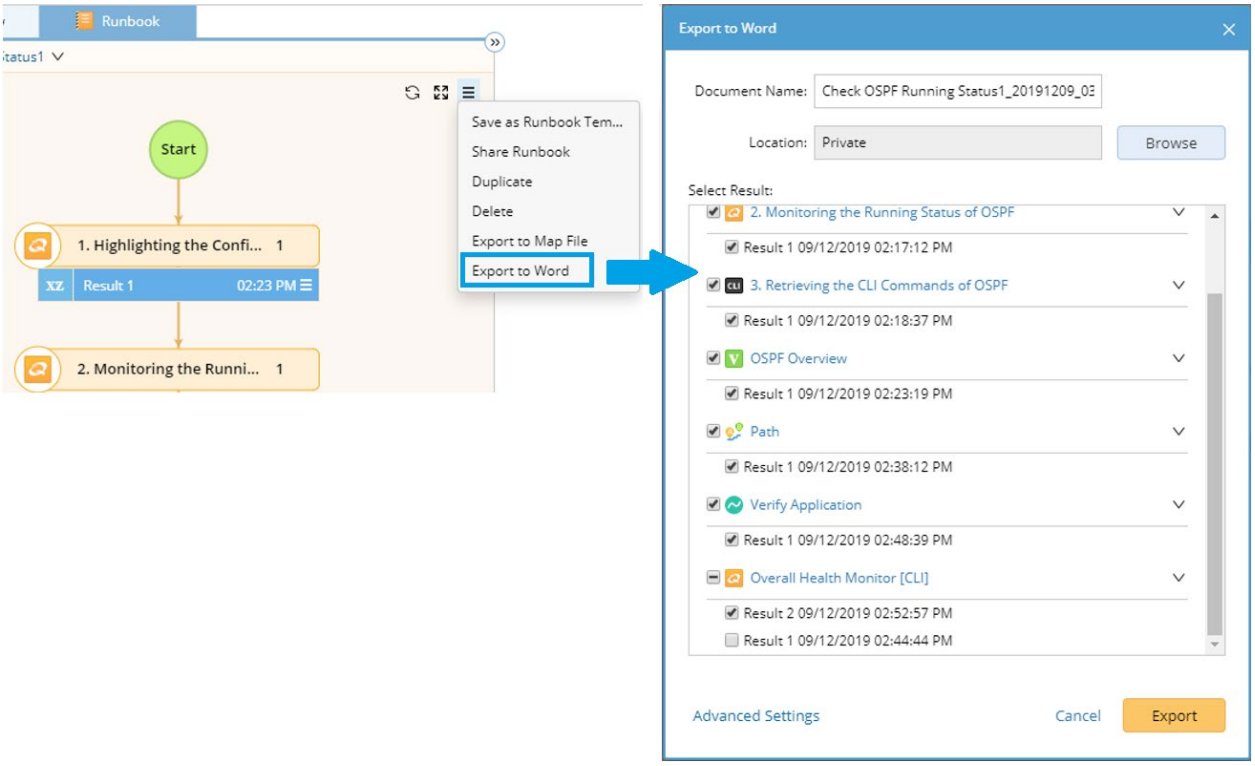
2.3.3.Gateway for Multicast Reverse Path

IEv8.02 uses the unicast path gateway logic to look up the gateway for a multicast path, but adds filter mechanism to the logic:

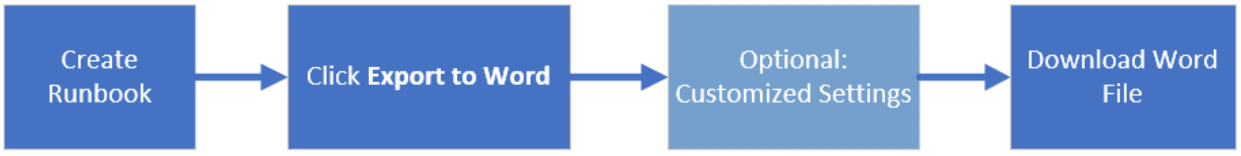
- If a candidate gateway is not configured with multicast, it will be removed from the gateway list.
- If devices associated with a particular technology such as HSRP, VXLAN anycast gateway, and VIP, is configured with multicast, they will be added to the gateway list.
- If the device that an input IP belongs to is a network device, the device will be added to the gateway list no matter it is configured with multicast or not.

2.4. Document Network Change and Runbook Automation in Word

Accurate documentation is critical for managing a reliable network. IEv8.02 adds the documentation feature for two important automation features: Change Management and Runbook. With the “Export to Word” function, each network change task can be documented in a controlled and consistent manner for archive and collaboration throughout the planning, approval, implementation and verification process, as well as other Runbook actions and results.



User Flow to Document Runbook



Check out the golden sample files exported from two Runbooks:

General Runbook

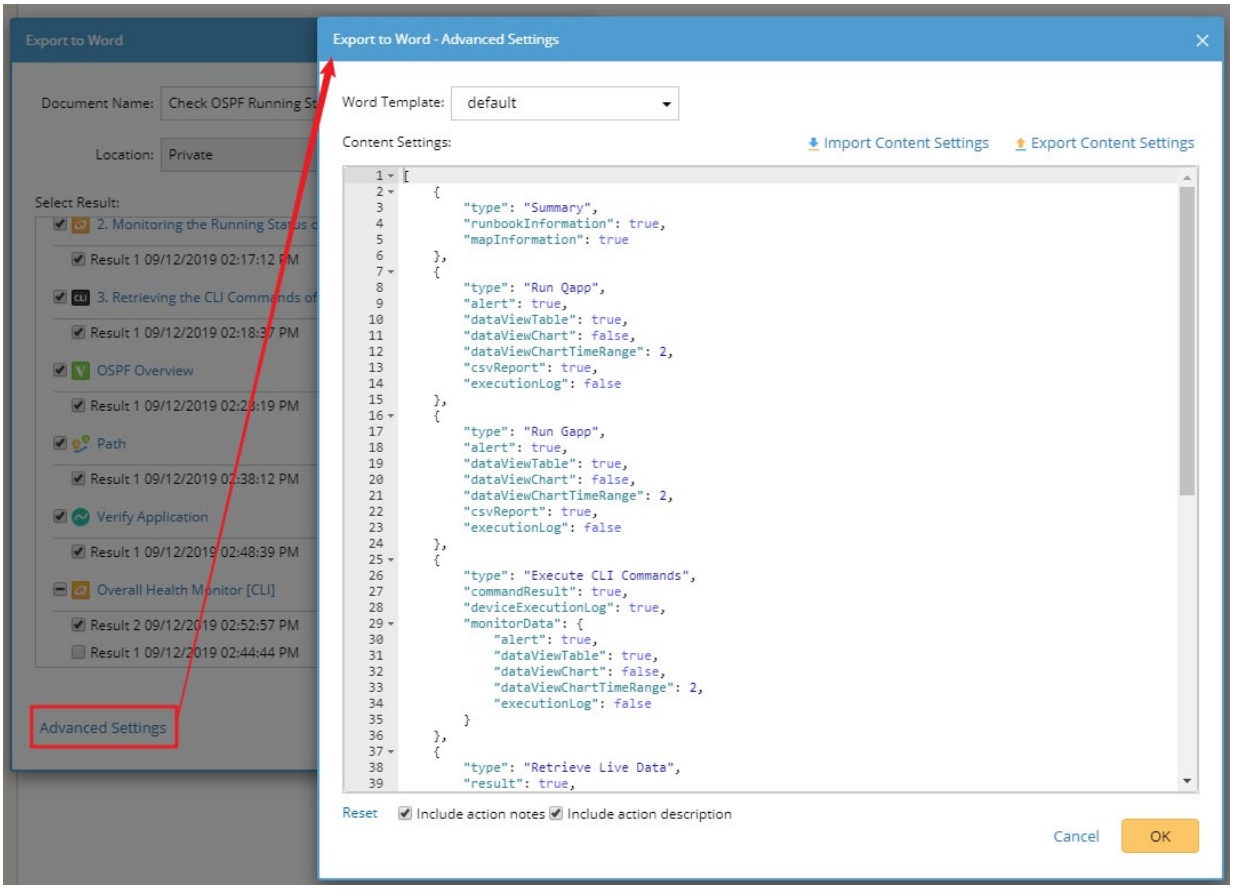
Check OSPF Running Status_201

Change Management Runbook

Change SNMP community string_

Custom Content Settings

Besides Runbook result selection, the Runbook content that can be exported is configurable in the Advanced Settings.



The default settings for the exported Runbook content can be referenced at online help.

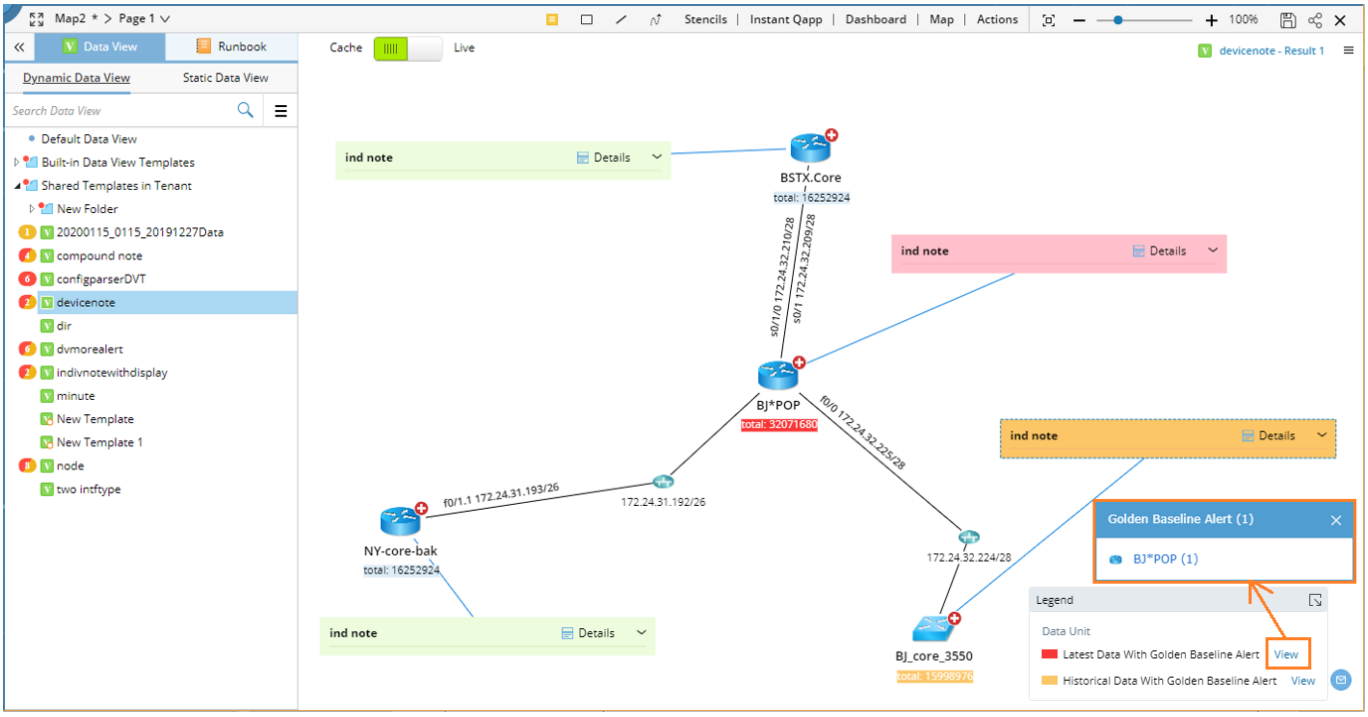
2.5. Enhancements to Reference Flow

2.5.1.New Indicators for Golden Baseline Alerts on Map

The Golden Baseline of a variable represents the reference standard of "healthy" status for a network metric type. Starting from IEv8.0, NetBrain allows the predefinition of the Golden Baseline for specific variables and can detect an anomaly as soon as it arises, for example, the value of `$CRC` increases or the value of `$eigrp_q_count` is not equal to 0. These anomalies can be highlighted on a map as alerts to warn users.

IEv8.02 adds new color and count indicators to visualize the hints for Golden Baseline Alerts proactively. With these new indicators in the Data View pane, end users can instantly know whether there are any recent or historical Golden Baseline Alerts per Data View Template, and how many of them. As a benefit, end users only need to apply a Data View Template that has an indicator to look into anomalies. From the map legend, end

users can quickly address the devices/interfaces that have an alert recently or historically.



Example of Indicator	Explanation
6 (Latest GB Alerts)	This data view has Golden Baseline Alerts for **6** device/interface variables, detected in the latest run. **Note:** If the latest run was one month ago, these alerts will be treated as Historical GB Alerts.
1 (Historical GB Alerts)	This data view has Golden Baseline Alerts for **1** device/interface variable, detected in the last six months.
6 (A Combination of Latest and Historical GB Alerts)	▪ This data view has Golden Baseline Alerts for **X** device/interface variables, detected in the latest run. ▪ This data view has Golden Baseline Alerts for **Y** device/interface variables, detected in the last six months. ▪ X + Y = 6

Note[1]**:** The GB alert count statistics only refer to those detected through scheduled DVT/parser tasks (cached data). The alerts detected by using live data are NOT included.

Note[2]**:** If two devices/interfaces in a data view contain the same variable that has an alert, the system calculates the alert count as 2.

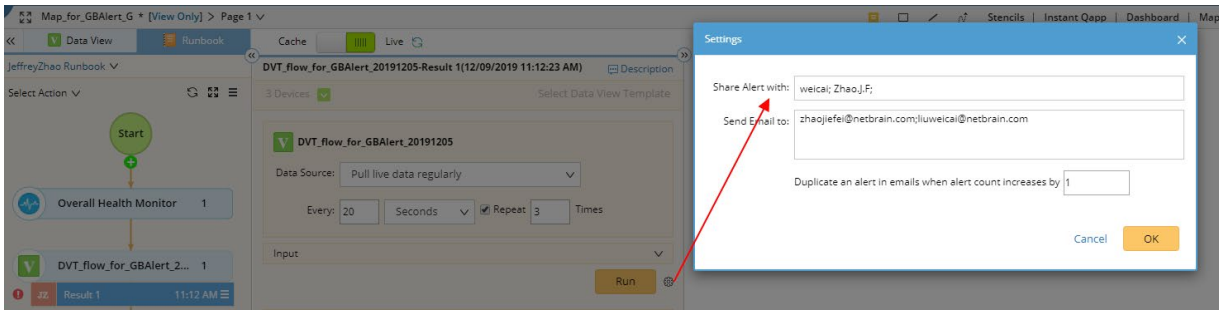
2.5.2.Message Golden Baseline Alerts to More Users

To warn users about network data deviations when they are not monitoring, IEv8.02 adds the function to message alerts via NetBrain system alerts, or email alerts, or both. As a result, end users can timely respond by analyzing the root cause on an addressed dynamic map and performing drill-down troubleshooting.

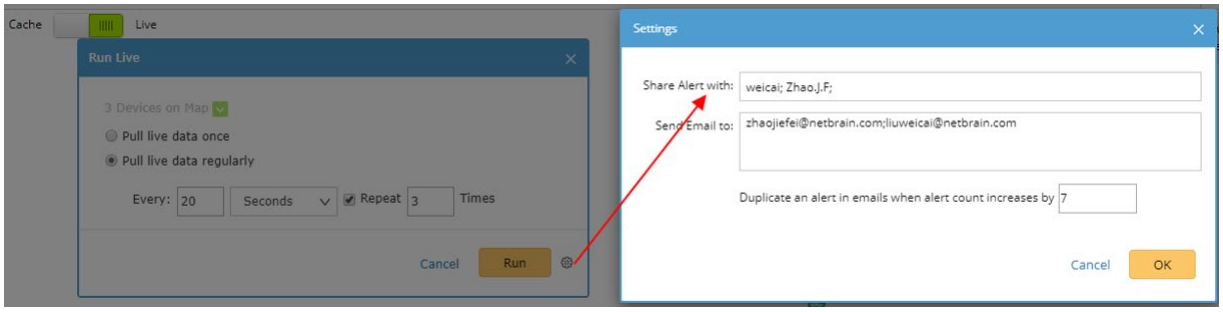
By default, only the action executor can receive system alerts for golden baseline deviations. The executor can configure whether to message more users proactively. Detected alerts will be categorized as errors in the Event Console.

Use Flow

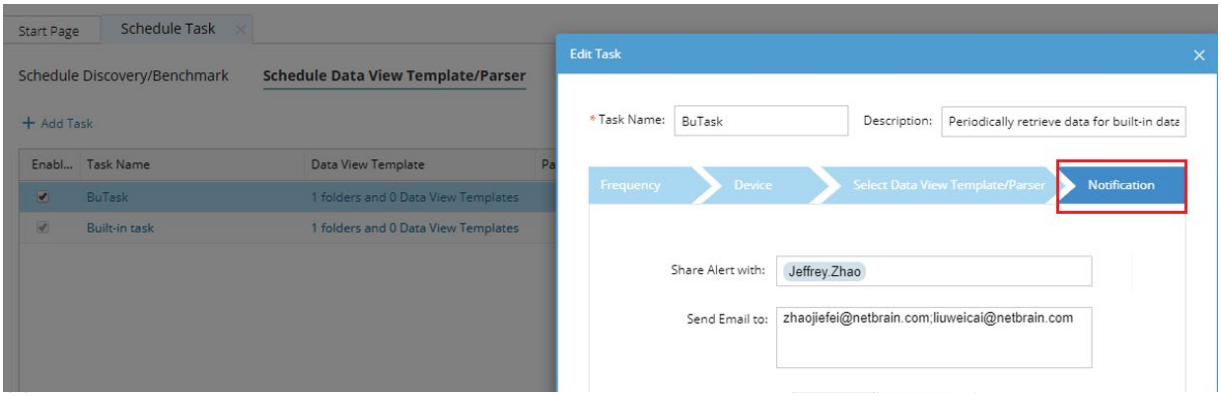
1. Configure the alert settings for a Data View Template by sharing with specified domain user accounts, or sending email alerts to specified email addresses, or both. For example:
   - Before running a Data View Template Node in a Runbook:



   - When toggling to apply a Data View Template with Live Data to a map:



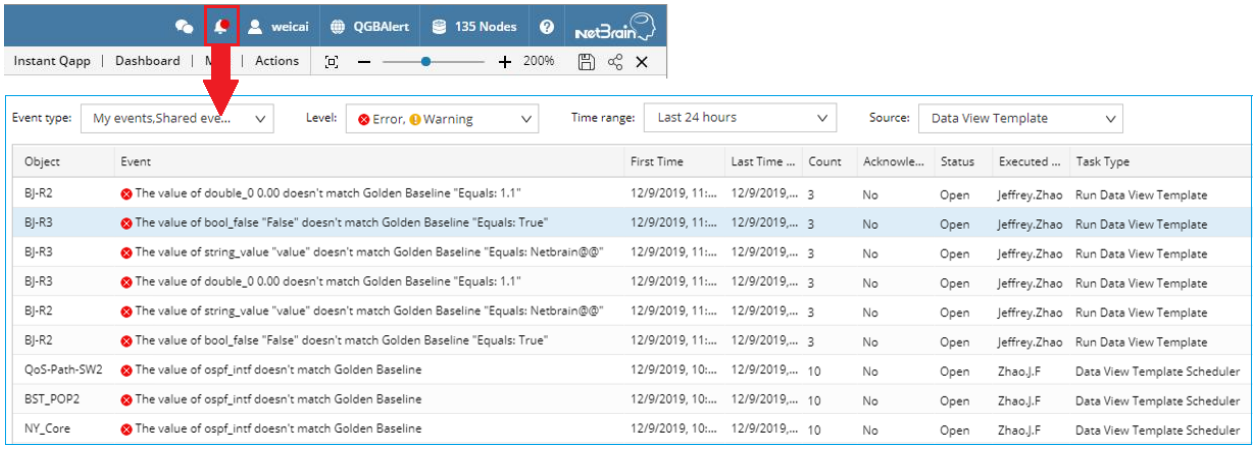
- When scheduling a Data View Template Task:



For more details about the email alerting rules, refer to Alerting Rules.

2. Receive alert messages from system alerts, or email alerts, or both.

Example 1: *Golden Baseline Alerts in System Event Console*



Object	Event	First Time	Last Time ...	Count	Acknowle...	Status	Executed ...	Task Type
BJ-R2	The value of double_0 0.00 doesn't match Golden Baseline "Equals: 1.1"	12/9/2019, 11:...	12/9/2019,...	3	No	Open	Jeffrey.Zhao	Run Data View Template
BJ-R3	The value of bool_false "False" doesn't match Golden Baseline "Equals: True"	12/9/2019, 11:...	12/9/2019,...	3	No	Open	Jeffrey.Zhao	Run Data View Template
BJ-R3	The value of string_value "value" doesn't match Golden Baseline "Equals: Netbrain@@"	12/9/2019, 11:...	12/9/2019,...	3	No	Open	Jeffrey.Zhao	Run Data View Template
BJ-R3	The value of double_0 0.00 doesn't match Golden Baseline "Equals: 1.1"	12/9/2019, 11:...	12/9/2019,...	3	No	Open	Jeffrey.Zhao	Run Data View Template
BJ-R2	The value of string_value "value" doesn't match Golden Baseline "Equals: Netbrain@@"	12/9/2019, 11:...	12/9/2019,...	3	No	Open	Jeffrey.Zhao	Run Data View Template
BJ-R2	The value of bool_false "False" doesn't match Golden Baseline "Equals: True"	12/9/2019, 11:...	12/9/2019,...	3	No	Open	Jeffrey.Zhao	Run Data View Template
QoS-Path-SW2	The value of ospf_intf doesn't match Golden Baseline	12/9/2019, 10:...	12/9/2019,...	10	No	Open	Zhao.J.F	Data View Template Scheduler
BST_POP2	The value of ospf_intf doesn't match Golden Baseline	12/9/2019, 10:...	12/9/2019,...	10	No	Open	Zhao.J.F	Data View Template Scheduler
NY_Core	The value of ospf_intf doesn't match Golden Baseline	12/9/2019, 10:...	12/9/2019,...	10	No	Open	Zhao.J.F	Data View Template Scheduler

Example 2: *Golden Baseline Alert Messages in Email*

17 Golden Baseline Alerts Detected for CDom_BRack_GBome_Map.Page 1.Result 3.DVT-Flow-for-GBAlert-20200111

qatest1@netbrain.com
To liuweicai@netbrain.com

Tenant: Initial Tenant
Domain: domain1
Task: CDom_BRack_GBome_Map.Page 1.Result 3.DVT-Flow-for-GBAlert-20200111 executed by user Zhao.J.F

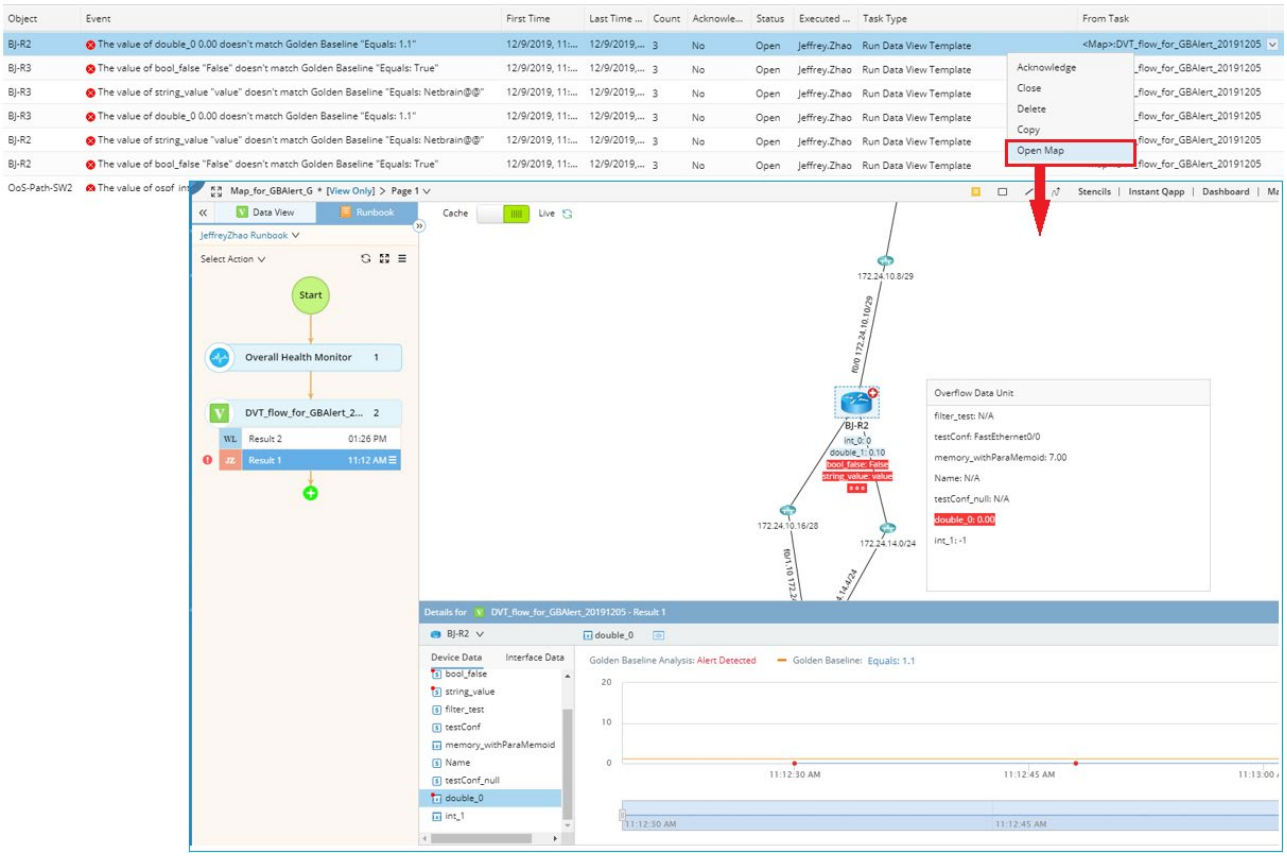
Golden Baseline Check for Device Variables:

Object	Message	Time
BJ*POP	The value of uptime_minutes 26 doesn't match Golden Baseline "Continuously Increase"	2020-01-17 14:54:51 +08:00
BJ_core_3550	The value of uptime_minutes 25 doesn't match Golden Baseline "Continuously Increase"	2020-01-17 14:54:51 +08:00
BJ_L2_Core_5	The value of table_test doesn't match Golden Baseline	2020-01-17 14:54:30 +08:00
BJ_L2_Core_5	The value of uptime_minutes 28 doesn't match Golden Baseline "Continuously Increase"	2020-01-17 14:54:51 +08:00
BJ_L2_Core_5	The value of table_test doesn't match Golden Baseline	2020-01-17 14:55:32 +08:00
BJ-L2-Core-A	The value of uptime_minutes 26 doesn't match Golden Baseline "Continuously Increase"	2020-01-17 14:55:21 +08:00

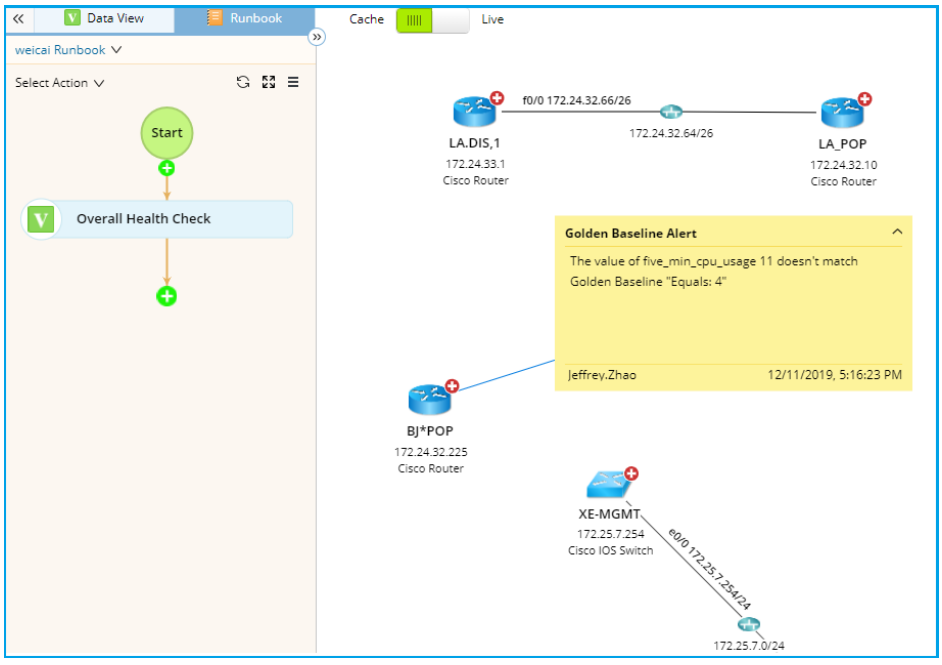
Golden Baseline Check for Interface Variables:

Object	Interface	Message	Time
BJ*POP	FastEthernet0/0	The value of input_drops 0 doesn't match Golden Baseline "Continuously Increase"	2020-01-17 14:54:51 +08:00
BJ*POP	Serial0/1/1	The value of input_drops 0 doesn't match Golden Baseline "Continuously Increase"	2020-01-17 14:54:51 +08:00
BJ_core_3550	FastEthernet0/1	The value of input_drops 0 doesn't match Golden Baseline "Continuously Increase"	2020-01-17 14:54:51 +08:00
BJ_core_3550	Port-channel10	The value of input_drops 0 doesn't match Golden Baseline "Continuously Increase"	2020-01-17 14:54:51 +08:00
BJ_L2_Core_5	Vlan10	The value of input_drops 0 doesn't match Golden Baseline "Continuously Increase"	2020-01-17 14:54:51 +08:00
BJ-L2-Core-A	Port-channel10	The value of input_drops 0 doesn't match Golden Baseline "Continuously Increase"	2020-01-17 14:54:51 +08:00
BJ-L2-Core-A	Vlan10	The value of input_drops 0 doesn't match Golden Baseline "Continuously Increase"	2020-01-17 14:54:51 +08:00
BST,POP1	Serial1	The value of input_error 9660 doesn't match Golden Baseline "Less or Equals: 7"	2020-01-17 14:54:31 +08:00
BST,POP1	Serial1	The value of input_error 9660 doesn't match Golden Baseline "Less or Equals: 7"	2020-01-17 14:54:51 +08:00
BST,POP1	Serial1	The value of input_drops 479 doesn't match Golden Baseline "Continuously Increase"	2020-01-17 14:54:51 +08:00
BST,POP1	Serial1	The value of input_error 9660 doesn't match Golden Baseline "Less or Equals: 7"	2020-01-17 14:55:31 +08:00

3. Open the map to troubleshoot.



Note: If the alert is detected in a scheduled DVT task, the alert message will be attached to the map as a device note.

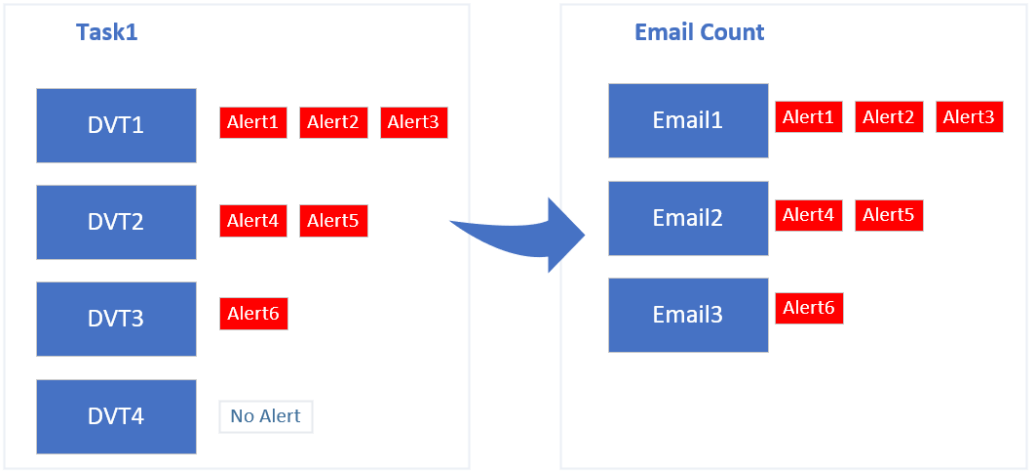


Emailing Rules for Golden Baseline Alerts

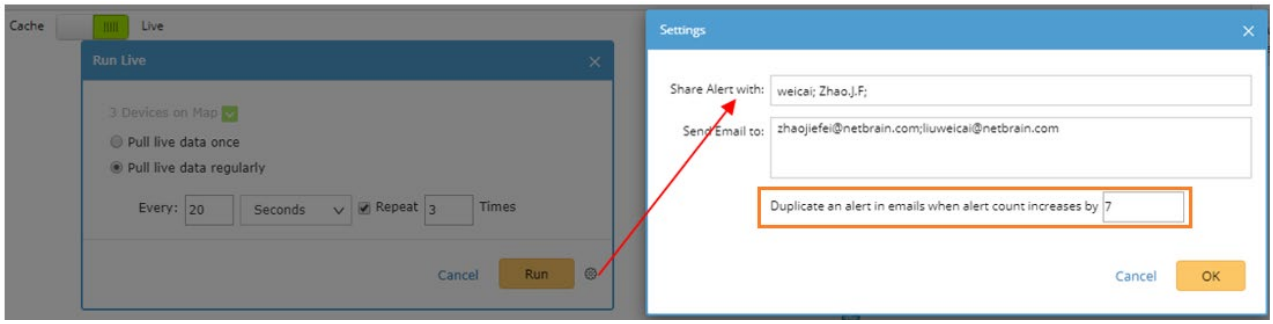
The system provides three general rules to avoid duplicate and excessive email alerts.

1. By default, the system sends alert emails every 5 minutes.
   The frequency is configurable at **System Management > Email Settings**.

2. The system emails GB Alerts per Data View Template.
   For example, if a task (on-demand or scheduled) contains 4 Data View Templates, and 3 of them have Golden Baseline deviations, the system will send out 3 emails separately.

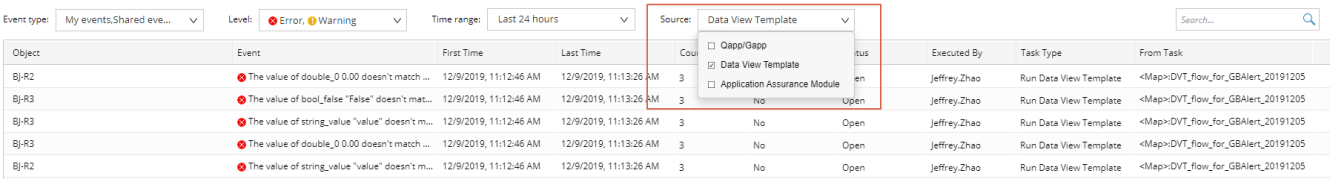


3. When the data source of an on-demand DVT task is "Pull live data regularly", the system will duplicate an alert message in one email if it reaches a specified threshold. It is configurable in the alert settings, and the default threshold value is 7.

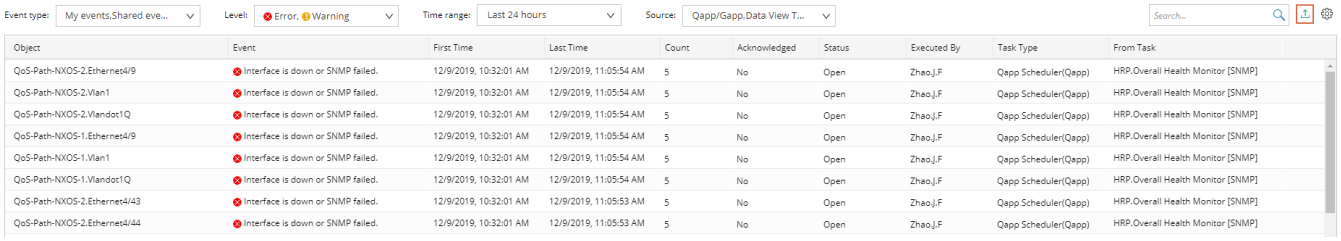


Minor Enhancements in Event Console

- Use three filters to narrow events by task source.

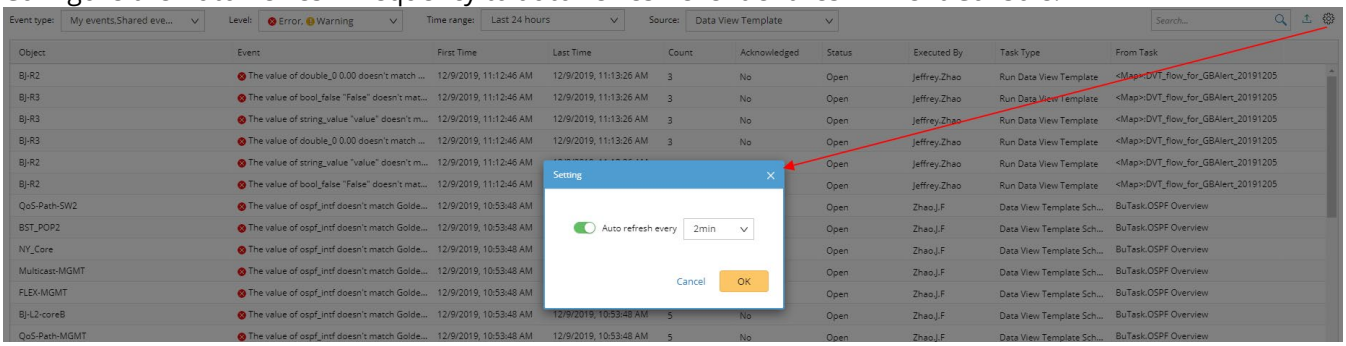


- Export events into a CSV file by clicking the Export button.

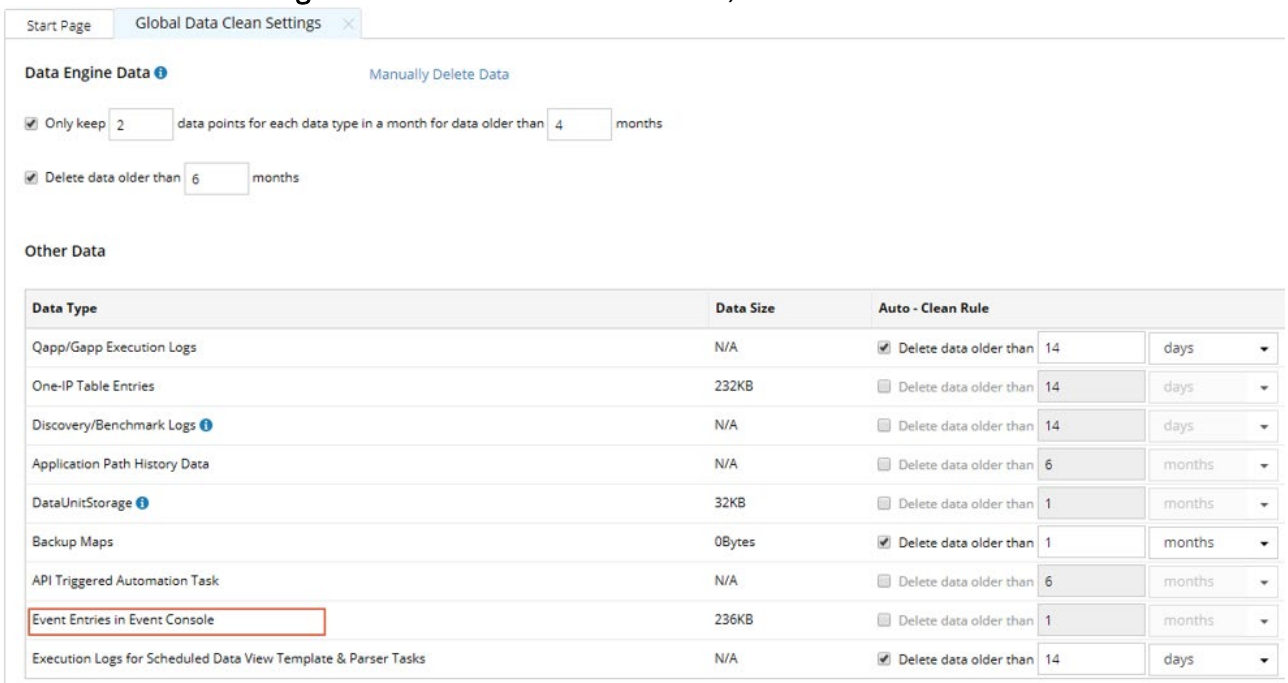


- Events are highlighted and differentiated through Error/Warning icons, rather than font colors.

- Configure the Auto Refresh Frequency to auto-refresh event entries in Event Console.



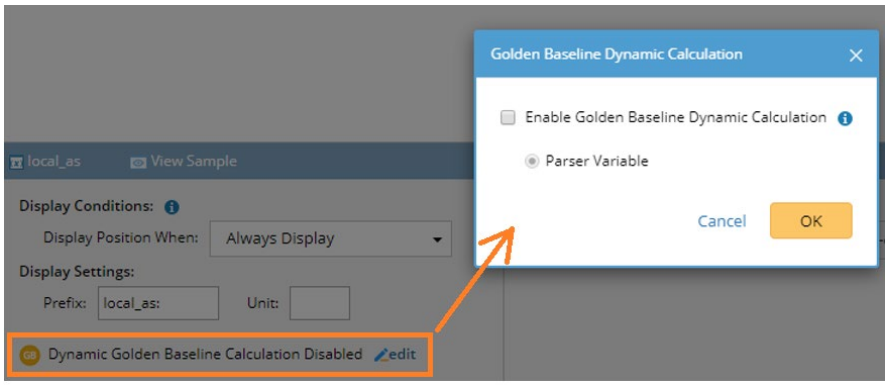
- Auto-clean event entries regularly by enabling and setting the corresponding rule (**Domain Management > Global Data Clean Settings > Event Entries in Event Console**).



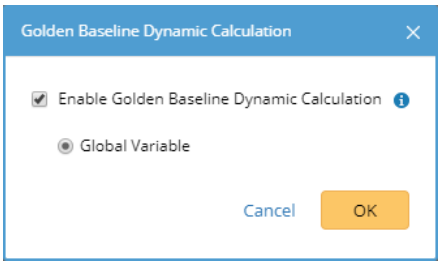
2.5.3.Enable Golden Baseline Dynamic Calculation for Parser Variable

In previous versions, users had to define a global variable first before enabling the dynamic calculation of the golden baseline (GB) for a parser variable.

IEv8.02 allows users to enable the GB dynamic calculation as soon as they place a parser variable of a legacy device to a device/interface position in a Data View Template.

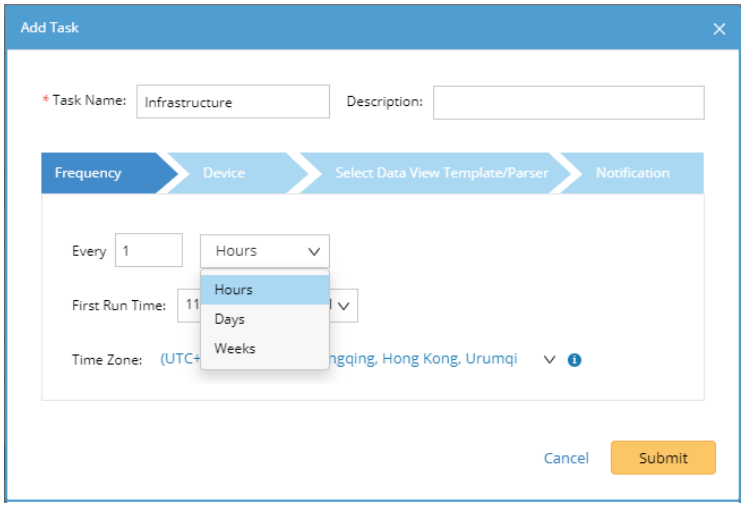


Note: The original enablement process based on each global variable is also available. If one parser variable has its global variable defined and associated, the dialog box will show as below:



2.5.4.Schedule Data View Template Task on an Hourly Basis

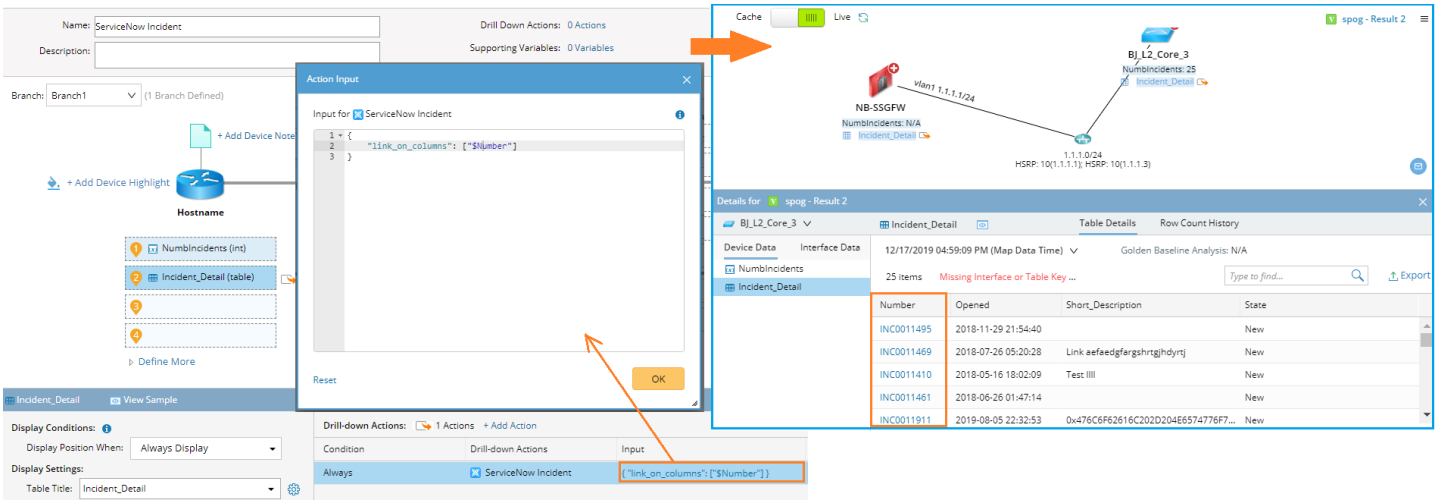
To support the schedule of a Data View Template Task at a higher frequency, IEv8.02 adds “Hours” as one of the unit options for the frequency setting. The default option is still “Days”.



2.5.5.Custom Column to Show SPOG URL in DVT

To enhance the reference flow for single pane of glass, IEv8.02 allows power users to attach a SPOG URL to a specific table column flexibly in a Data View Template.

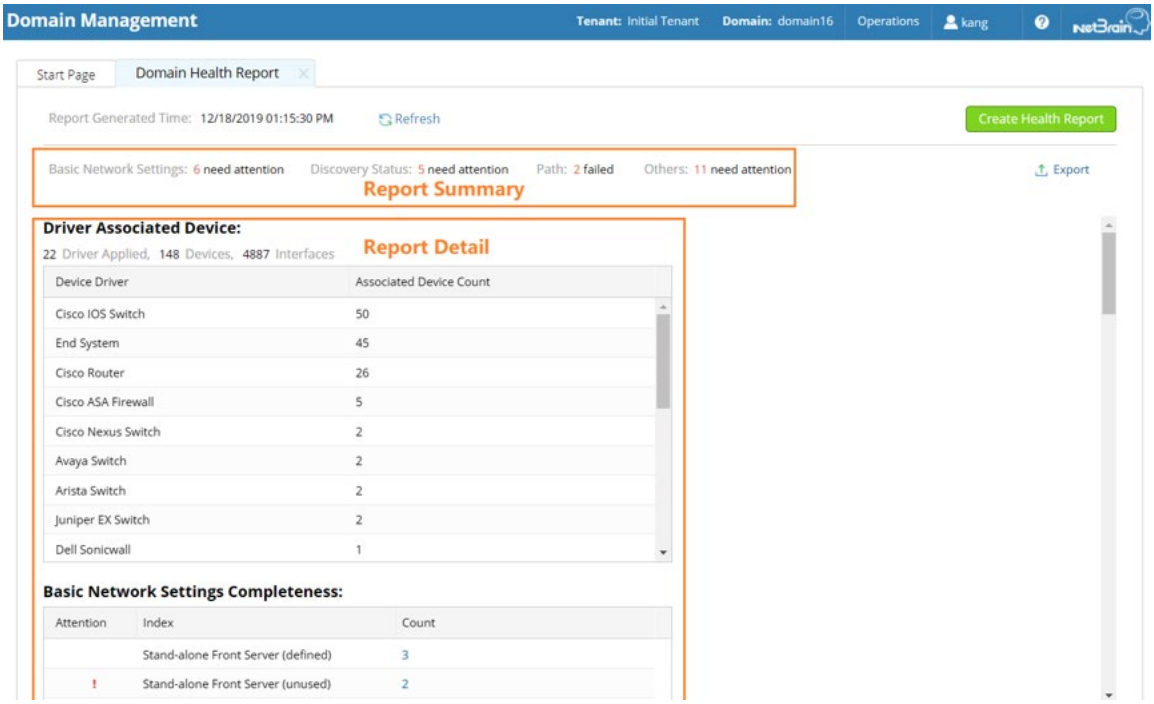
For example, to attach the ServiceNow Incident URL to the Incident Numer column of the table `Incident_Detail`, users can assign the `$Number` variable as the value of the `link_on_columns` parameter in the script of Action Input. As a benefit, when end users look into the table of incident details on a map, they can be redirected to the ServiceNow system.



2.6. New Domain Health Report

To accelerate the tune-up phase (customization for topology/path data accuracy) before delivering a fine-tuned domain to end users, IEv8.02 provides a report with more detailed statistics about domain health. As a benefit, the domain administrators can proactively focus on the items that need their attention and resolve the potential problems existing in a domain as earlier as possible.

The domain health report is generated on-demand and contains both a summary and a variety of categories.



Report Category	Description and Example
Summary	This summary shows the number of items that need the domain admin to pay attention to.
Driver Associated Device	This category shows the number of devices that are discovered by drivers. 22 Driver Applied, 148 Devices, 4887 Interfaces Device Driver / Associated Device Count Cisco IOS Switch 50 End System 45 Cisco Router 26 Cisco ASA Firewall 5 Cisco Nexus Switch 2 Avaya Switch 2 Arista Switch 2 Juniper EX Switch 2 Dell Sonicwall 1
Basic Network Setting Completeness	This category shows the summary of the predefined Network Settings, including the number of Front Server, Front Server Group, Private Key, Jumpbox, Telnet/SSH Login, Privilege Login, SNMP

<table>
<tr><th>Report Category</th><th>Description and Example</th></tr>
<tr><td></td><td>String, API Server, CheckPoint OPSEC and their usage.
<table>
<tr><th>Attention</th><th>Index</th><th>Count</th></tr>
<tr><td></td><td>Stand-alone Front Server (defined)</td><td>3</td></tr>
<tr><td>!</td><td>Stand-alone Front Server (unused)</td><td>2</td></tr>
<tr><td></td><td>Stand-alone Front Server (over 5000 devices)</td><td>0</td></tr>
<tr><td></td><td>Front Server Group (defined)</td><td>0</td></tr>
<tr><td></td><td>Front Server Group (unused)</td><td>0</td></tr>
<tr><td></td><td>Front Server Group (over 5000 devices per ...</td><td>0</td></tr>
<tr><td></td><td>Private Key (defined)</td><td>6</td></tr>
<tr><td>!</td><td>Private Key (unused)</td><td>6</td></tr>
<tr><td></td><td>Jumpbox (defined)</td><td>3</td></tr>
<tr><td>!</td><td>Jumpbox (unused)</td><td>3</td></tr>
<tr><td></td><td>Telnet/SSH Login (defined)</td><td>44</td></tr>
<tr><td>!</td><td>Telnet/SSH Login (unused)</td><td>44</td></tr>
<tr><td></td><td>Privilege Login (defined)</td><td>21</td></tr>
<tr><td>!</td><td>Privilege Login (unused)</td><td>21</td></tr>
<tr><td></td><td>SNMP String (defined)</td><td>18</td></tr>
<tr><td>!</td><td>SNMP String (unused)</td><td>18</td></tr>
<tr><td></td><td>API Server</td><td>11</td></tr>
<tr><td></td><td>CheckPoint OPSEC</td><td>0</td></tr>
</table>

Note: The attention (!) is a reminder, but not an alert. If the configuration is not necessary, users can ignore the attention.</td></tr>
<tr><td>Discovery Status</td><td>This category shows a summary of the discovery report, license usage, and issues encountered in the discovery process.
<table>
<tr><th>Attention</th><th>Index</th><th>Count</th></tr>
<tr><td></td><td>Discovered Devices</td><td>Legacy Device: 102</td></tr>
<tr><td></td><td>Licensed Node Usage</td><td>102 out of 10000 Licensed Nodes (Usage: 1.02%)</td></tr>
<tr><td>!</td><td>Discovered by SNMP</td><td>81</td></tr>
<tr><td>!</td><td>Unknown IP</td><td>82</td></tr>
<tr><td></td><td>Missed Devices</td><td>0</td></tr>
<tr><td>!</td><td>Unclassified Network Devices</td><td>1</td></tr>
<tr><td>!</td><td>Unknown SNMP SysObjectID</td><td>1</td></tr>
<tr><td>!</td><td>Subnet with Conflicted IPs</td><td>5</td></tr>
<tr><td></td><td>Zone</td><td>4</td></tr>
<tr><td></td><td>Hostname Change</td><td>0</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Report Category</th><th>Description and Example</th></tr>
<tr><td>Site Definition Completeness</td><td>This category shows the usage summary of each site.
Latest Build Time: 12/18/2019 01:15:30 PM
<table><tr><th>Attention</th><th>Index</th><th>Count</th></tr><tr><td>!</td><td>Container Site</td><td>0</td></tr><tr><td>!</td><td>Leaf Site</td><td>0</td></tr><tr><td></td><td>Leaf Site (with 0 device)</td><td>0</td></tr><tr><td></td><td>Leaf Site (over 100 devices)</td><td>0</td></tr><tr><td>!</td><td>Unassigned Devices</td><td>148</td></tr></table></td></tr>
<tr><td>Benchmark Task Health</td><td>This category shows the latest two execution summaries of each benchmark task.
Basic System Benchmark
<table><tr><th>Attention</th><th>Execution Time</th><th>Result</th><th>Duration</th><th>Configuration Retrieval Success Rate</th></tr><tr><td></td><td>12/16/2019 12:58:31 PM</td><td>Succeeded</td><td>8 mins 34 secs</td><td>99.32% (145/146 devices)</td></tr></table>Benchmark1
<table><tr><th>Attention</th><th>Execution Time</th><th>Result</th><th>Duration</th><th>Configuration Retrieval Success Rate</th></tr><tr><td></td><td>12/18/2019 9:39:13 AM</td><td>Succeeded</td><td>37 secs</td><td>40% (2/5 devices)</td></tr></table></td></tr>
<tr><td>Path Calculation Health</td><td>This category shows the number of paths that have calculation failures.
Total 5 paths, 3 succeeded, 0 not executed, 2 failed</td></tr>
<tr><td>Disk Management Setting Completeness</td><td>This category shows the summary of the predefined auto-clean rules for global data and the disk alert rules for MongoDB.
Global Data Clean Settings
<table><tr><th>Attention</th><th>Data Type</th><th>Data Clean</th></tr><tr><td></td><td>Data Engine Data</td><td>Enabled</td></tr><tr><td></td><td>Qapp/Gapp Execution Logs</td><td>Enabled</td></tr><tr><td>!</td><td>One-IP Table Entries</td><td>Disabled</td></tr><tr><td>!</td><td>Discovery/Benchmark Logs</td><td>Disabled</td></tr><tr><td>!</td><td>Application Path History Data</td><td>Disabled</td></tr><tr><td>!</td><td>DataUnitStorage</td><td>Disabled</td></tr><tr><td></td><td>Backup Maps</td><td>Enabled</td></tr><tr><td>!</td><td>API Triggered Automation Task</td><td>Disabled</td></tr><tr><td>!</td><td>Event Entries in Event Console</td><td>Disabled</td></tr><tr><td></td><td>Execution Logs for Scheduled Data View Template & Parser Tasks</td><td>Enabled</td></tr><tr><td></td><td>Dashboard Activity Data Table</td><td>Enabled</td></tr></table>MongoDB Disk Alert Rules: (System Settings)
<table><tr><th>Attention</th><th>Type</th><th>Status</th></tr><tr><td>!</td><td>Send emails when alert detected</td><td>Disabled</td></tr><tr><td>!</td><td>Send emails and delete Data Engine data when alert detected</td><td>Disabled</td></tr><tr><td></td><td>Send emails and disable write permisssion to MongoDB when alert detected</td><td>Enabled</td></tr><tr><td>!</td><td>Email Address</td><td>None</td></tr></table></td></tr>
</table>

Report Category	Description and Example
Map Layout Settings Completeness	This category shows the number of custom tags.

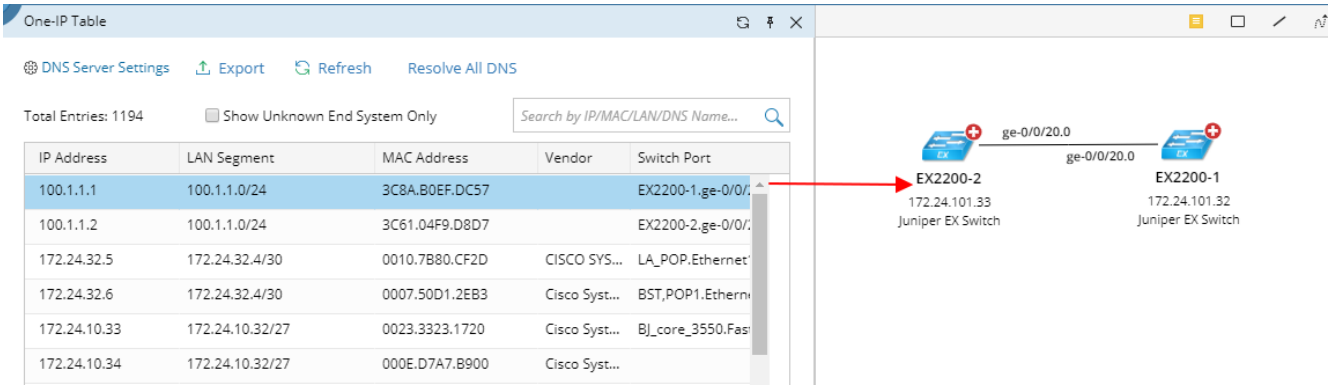
Attention	Index	Count
!	Customized Layout	0
	Devices with Tags Associated	0
!	Devices without Tags Associated	148
	Site Maps with Layouts Associated	0
	Site Maps without Layouts Associated	0

For more reference about the report content, refer to online help.

2.7. Map Enhancements

2.7.1.Map One-IP Table Entries with Drag-and-Drop

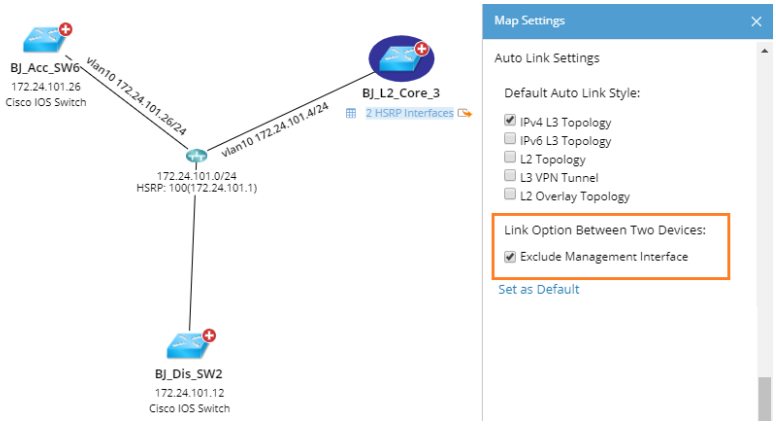
Launching the One-IP Table from the start menu will open a pane rather than a separate tab, so that users can map out devices and their L2 topology links with easy drag-and-drop, or compare table entries with the topology on a map.



2.7.2.Narrow Down Auto-link Scope and Count

Enable to Exclude Management Links from a Map

IEv8.02 allows users to exclude management links from a map when using the auto-link function. By default, this option is enabled.



Note: Considering management links are usually managed in one subnet, IEv8.02 provides a new built-in plugin "**recognize_management_interface**" to identify management links by subnet matching. This plugin is executed along with the Basic System Benchmark. Those interfaces within the specified subnets will be identified as management interfaces, and their interface property `isMgmtIntf` will be updated to "true" in the GDR.

Limit the Number of One-Time Auto Links

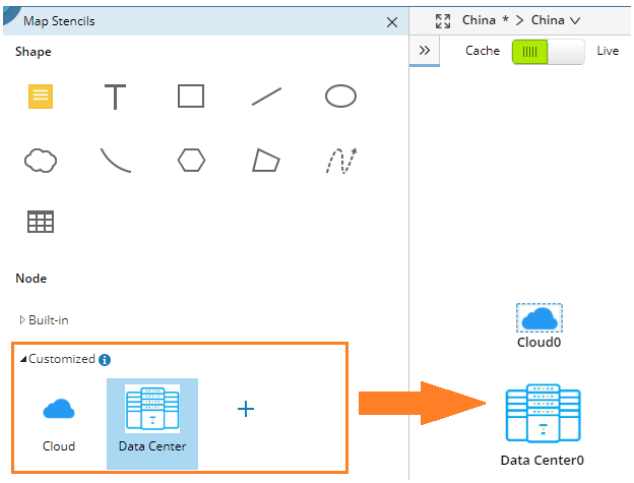
To improve the performance for the auto-link function, IEv8.02 adds control to limit that at most 50 devices can be auto-linked at one time on a map. Users have to repeat the auto-link action to link more devices.

Tip: This threshold is configurable in back-end config files.

2.7.3.Custom Node Icons

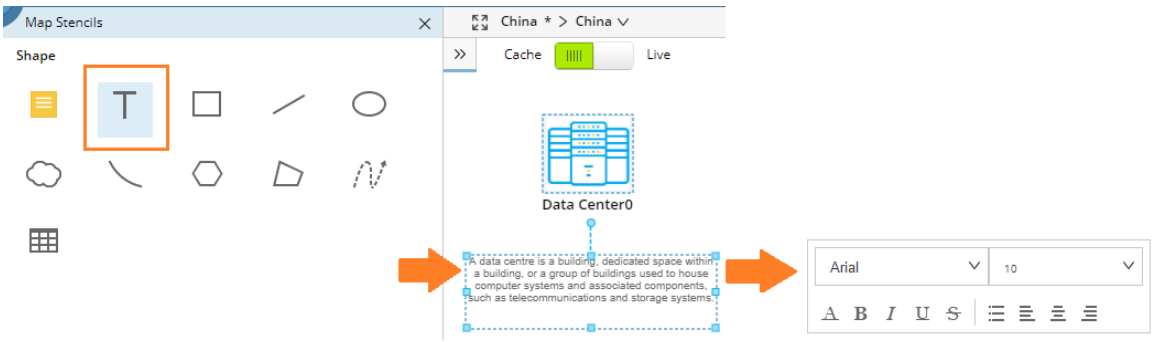
To meet the diverse needs for network mapping, IEv8.02 allows users to upload custom node icons to the Map Stencils pane. The uploaded node icons can be directly used for mapping with drag-and-drop.

Note: Only PNG is allowed.



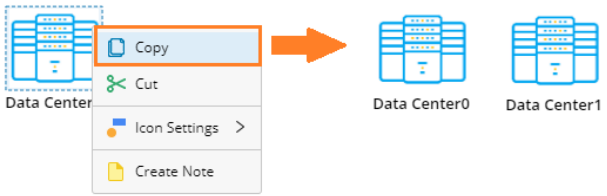
2.7.4.Free Text for Network Design/Troubleshooting Annotation

IEv8.02 adds the shape of Text to the Map Stencil pane. The style of the text can be customized, such as font, size, color, alignment, etc.



2.7.5.Copy, Cut and Paste Stencil Icons and Shapes

IEv8.02 adds the copy, cut, and paste functions for map components, including icons and shapes.

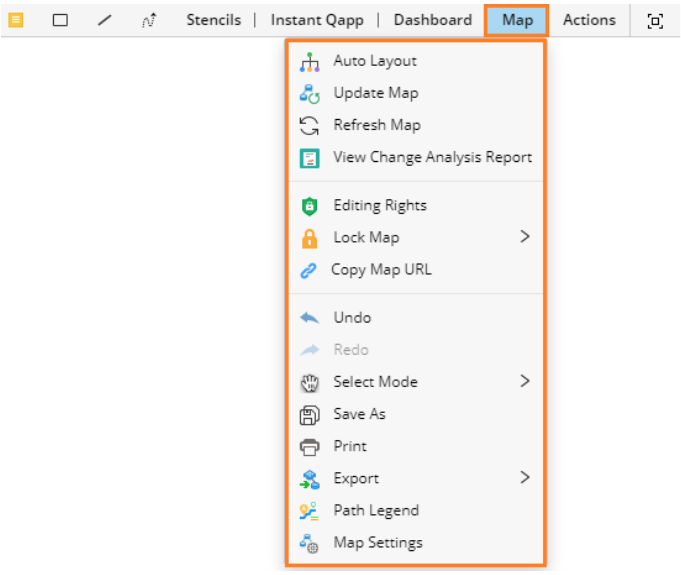


The following shortcut keys are allowed:

- **Copy** (CTL+C)
- **Cut** (CTL+X)
- **Paste** (CTL+V)

2.7.6.Enriched Map Operation Menu

IEv8.02 adds a variety of map tools to the map operation menu and optimizes the menu order.

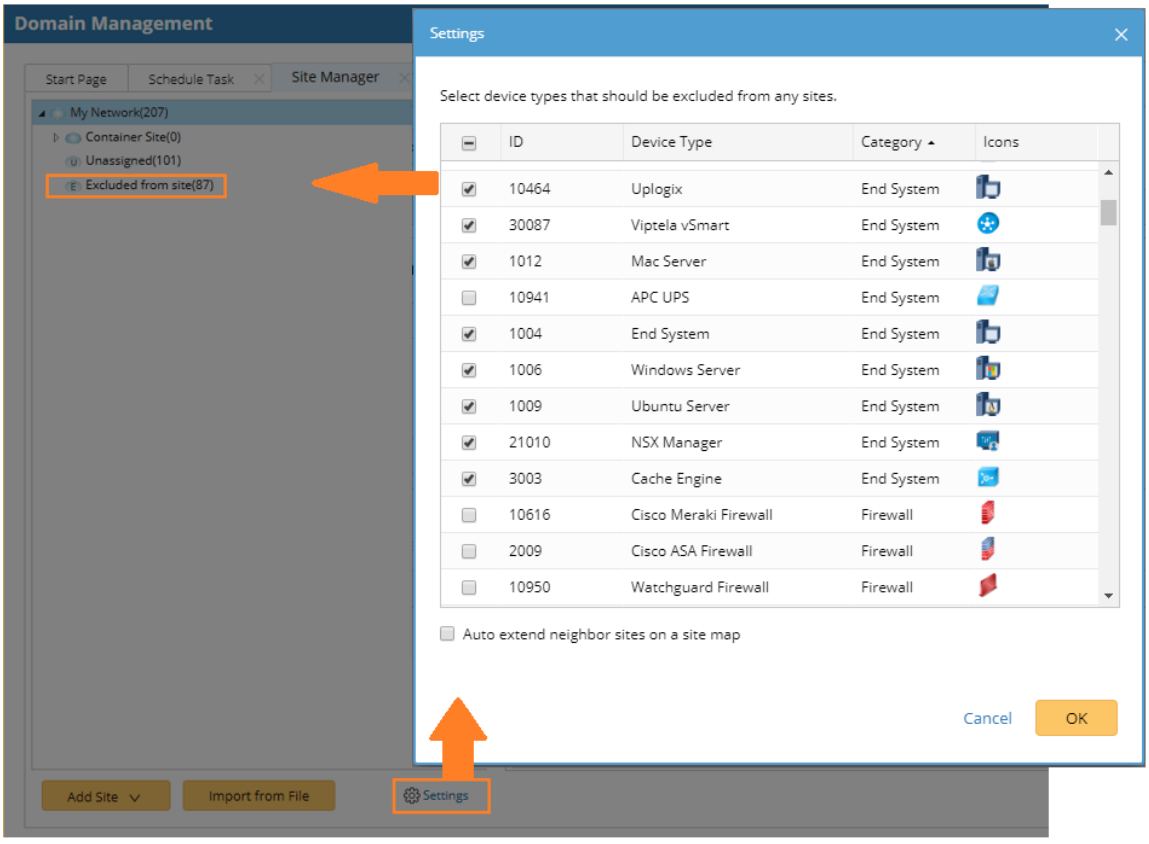


2.8. Site Enhancements

2.8.1.Exclude Specified Device Types from Site Assignment

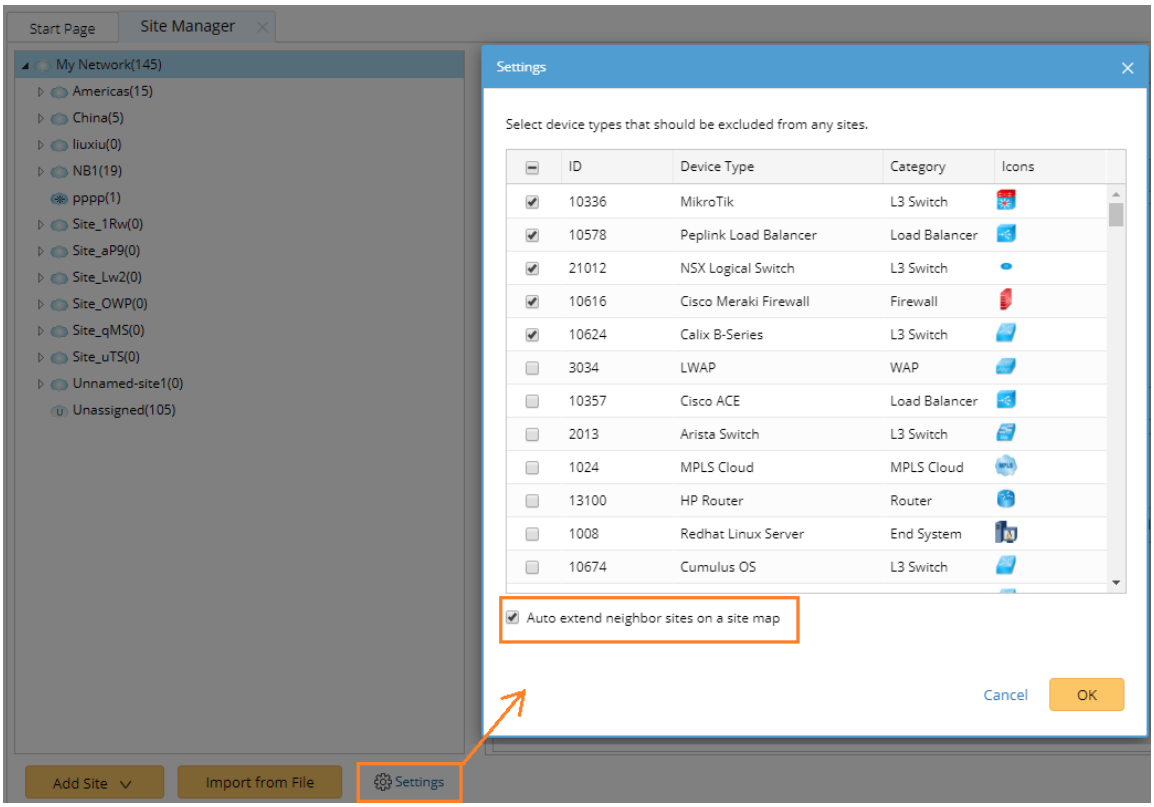
During a domain setup process, one of the power users' tasks is to complete site management by assigning many "unassigned" devices to target sites they belong to. However, in real-world cases, many End Systems and WAPs, such as PCs and printers, are not involved in any site build or topology build. It is usually time-consuming to go through the long "unassigned" list to get rid of them.

To elevate the operation efficiency of site management, IEv8.02 allows power users to exclude specific device types from the scope at the beginning. With this setting configured, devices of the specified devices types are categorized into a new category "Excluded from site" in the site tree and cannot be assigned to any site.



2.8.2.Allow Hiding Neighbor Sites from a Site Map

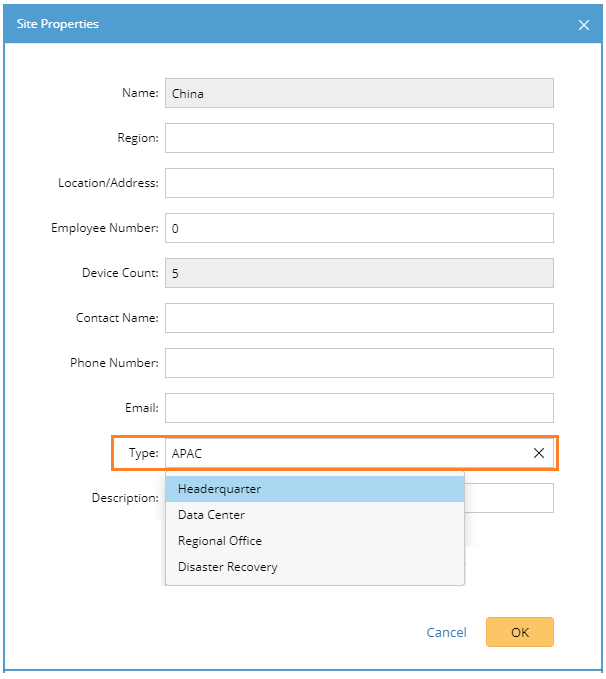
IEv8.02 adds control to hide neighbor sites from a site map to avoid mapping unnecessary neighbors. By default, this function is enabled.



To unhide neighbor sites, go to **Site Manager > Settings** and select the **Auto extend neighbor sites on a site map** check box.

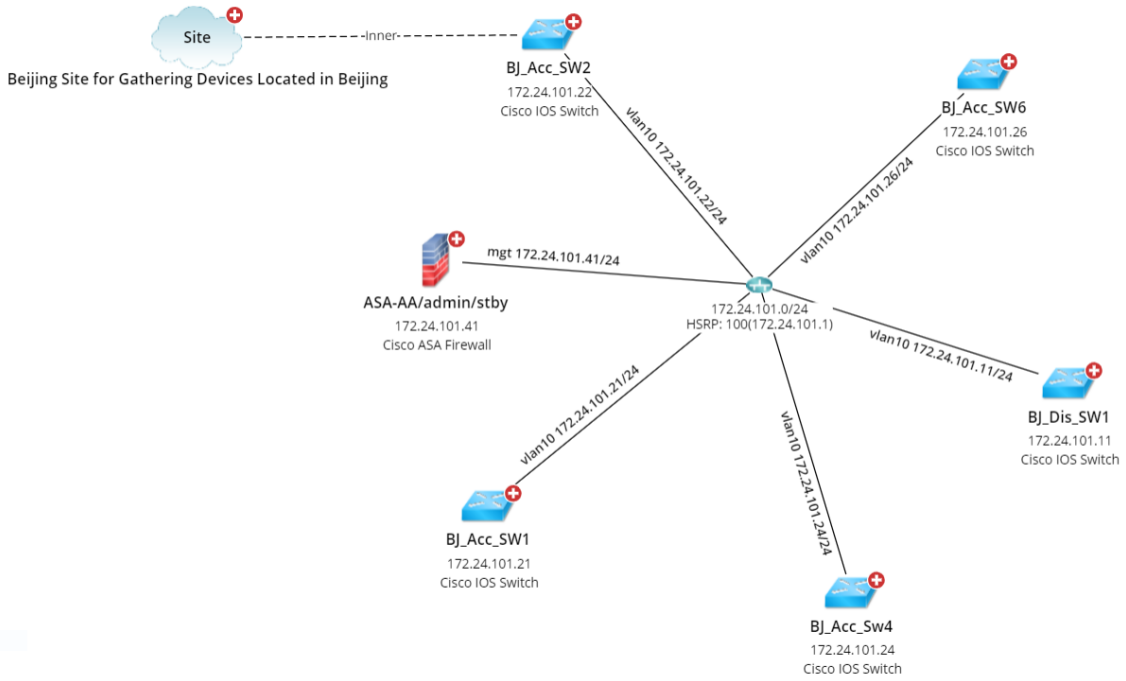
2.8.3.Allow User Input for Site Type

The “Site Type” property is used to label the usage of a site, such as headquarter, data center, regional office, or disaster recovery. Besides these built-in options, IEv8.02 allows custom user input to support more definitions, such as by location, by network architecture or by branch.



2.8.4.Move Site Name Underneath Site Icon to Show Full Site Name

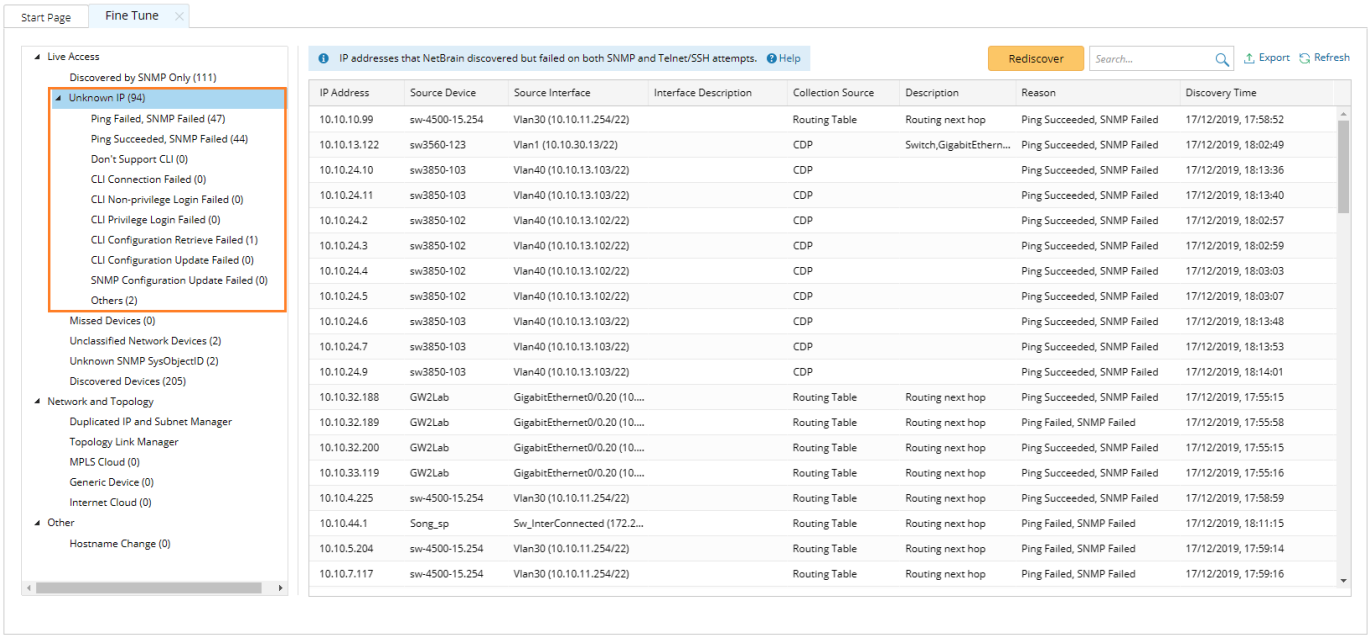
To fully display a site name on a map despite the scale of a site icon, IEv8.02 changes the visual design of the site icon by moving the site name underneath the site icon.



2.9. Enhancements to Domain Maintenance

2.9.1.Categorize Unknown IP by Reason

IEv8.02 splits 10 sub-categories for the Unknown IP category in the Fine Tune module to sort discovered IP addresses by reason.

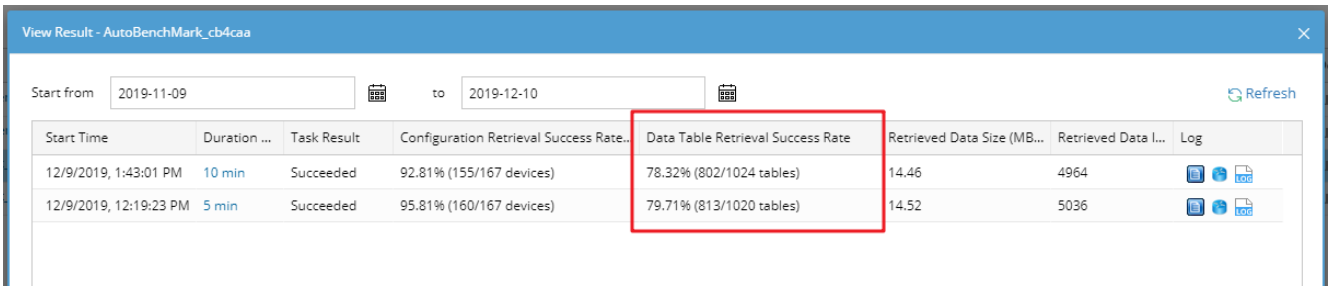


Category (by Reason)	Description
Ping Failed, SNMP Failed	It contains the discovered IP addresses that the system failed to ping or access via SNMP.
Ping Succeeded, SNMP Failed	It contains the discovered IP addresses that the system can ping successfully but cannot access via SNMP.
Don't Support CLI	It contains the discovered IP addresses that don't support Telnet/SSH access.
CLI Connection Failed	It contains the discovered IP addresses that the system failed to access via both Telnet/SSH and SNMP.
CLI Non-Privilege Login Failed	It contains the discovered IP addresses that the system can access via Telnet/SSH but failed to log in.
CLI Privilege Login Failed	It contains the discovered IP addresses that the system can log in via Telnet/SSH in Non-privilege mode, but failed in Privilege mode.
CLI Configuration Retrieval Failed	It contains the discovered IP addresses that the system can log in via Telnet/SSH but failed to retrieve CLI configurations.
CLI Configuration Update Failed	It contains the discovered IP addresses that the system can log in via Telnet/SSH but failed to update the retrieved CLI configurations.
SNMP Configuration Update Failed	It contains the discovered IP addresses that the system can access via SNMP but failed to access via Telnet/SSH and update SNMP configurations fully.
Others	None of the above.

For more details, refer to online help.

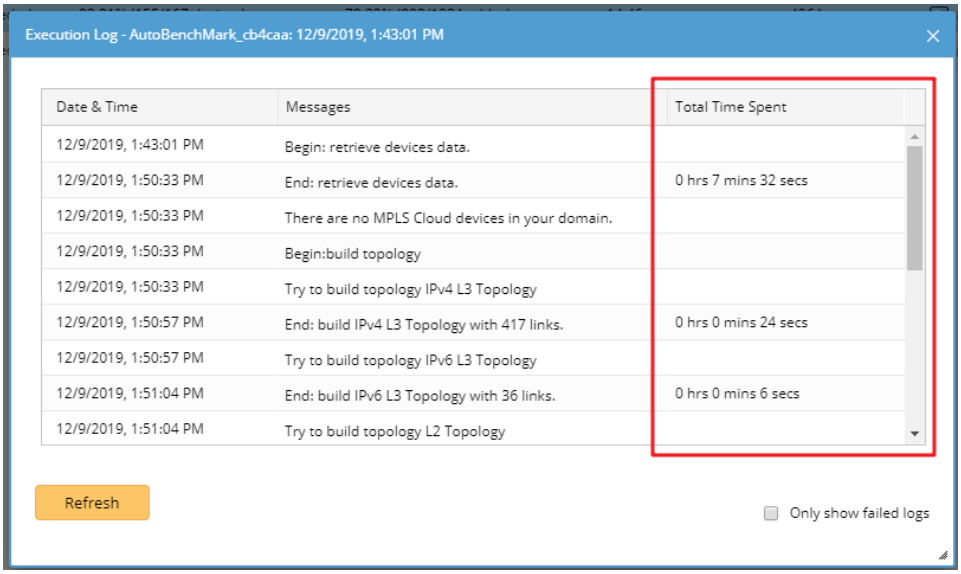
2.9.2.Show Data Table Retrieval Success Rate in Benchmark Report

To provide a hint for path analysis, IEv8.02 adds a column for the retrieval success rate of data tables to each benchmark result, including system tables and network control tables (NCT).



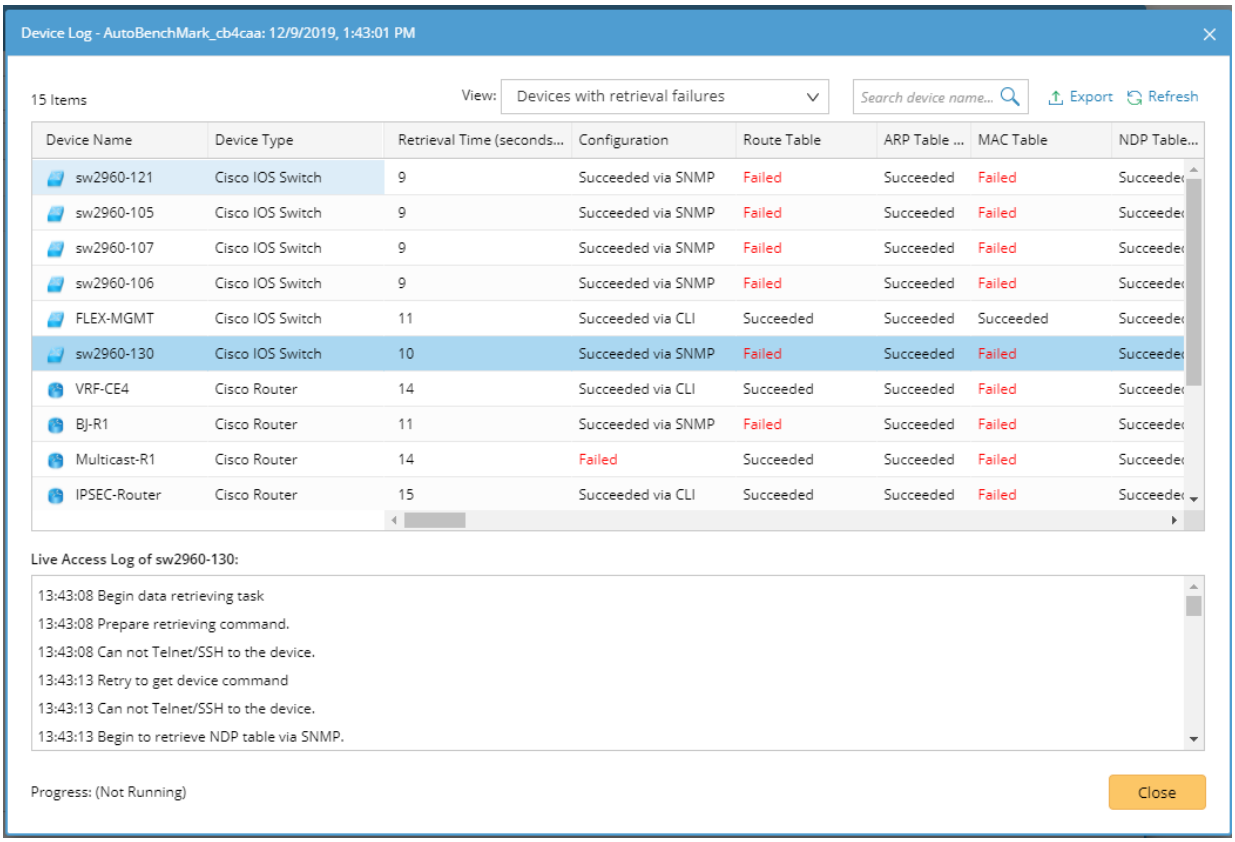
2.9.3.Enhanced Discovery/Benchmark Execution Log

To enable the lookup for the most time-consuming sub-task in a discovery/benchmark task, IEv8.02 separates a new column for Total Time Spent in the task execution log. This column shows the time spent on each sub-task and can be sorted by value size.



2.9.4.Enhanced Benchmark Device Log

IEv8.02 introduces the following usability enhancements to the benchmark device log:



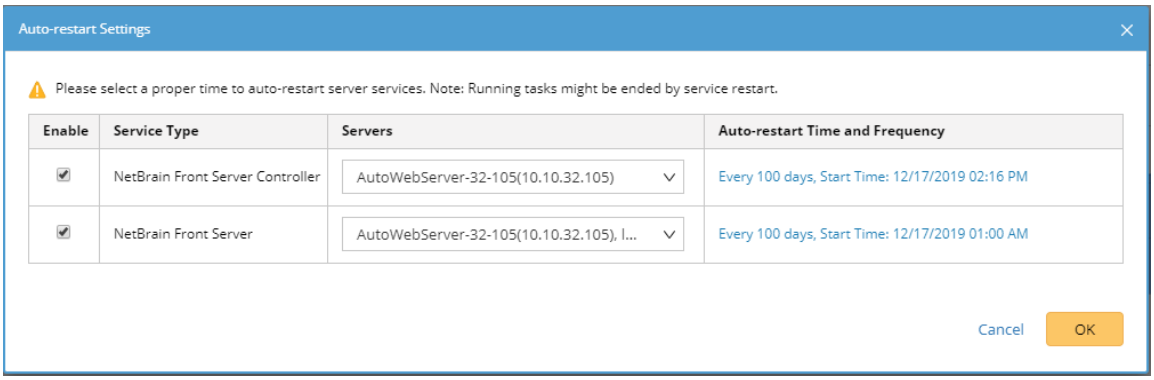
- Add device icons in the column of Device Name and add a column for Device Type to distinguish the vendor/model.
- Freeze the two columns: Device Name and Device Type to ensure users won't be lost when scrolling horizontally.
- Change the font color for failures to red.

2.10. Enhancements to Service Monitor

2.10.1. Schedule Service Auto-Restart for Front Server (Controller)

The longer time the system has been running for, the more memory usage it will consume, and the more junk data it will generate. IEv8.02 adds the capability to schedule an auto-restart of the service for Front Server and Front Server Controller recurringly, to solve some complicated issues with can be fixed by a restart button.

By default, the auto-restart settings are disabled. Users can enable an auto-restart on specified servers and select a proper time point and frequency.

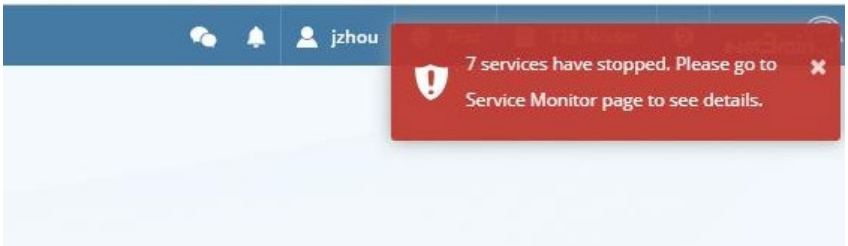


Note: Running tasks might be ended by a service restart.

2.10.2. Alert for Stopped Services

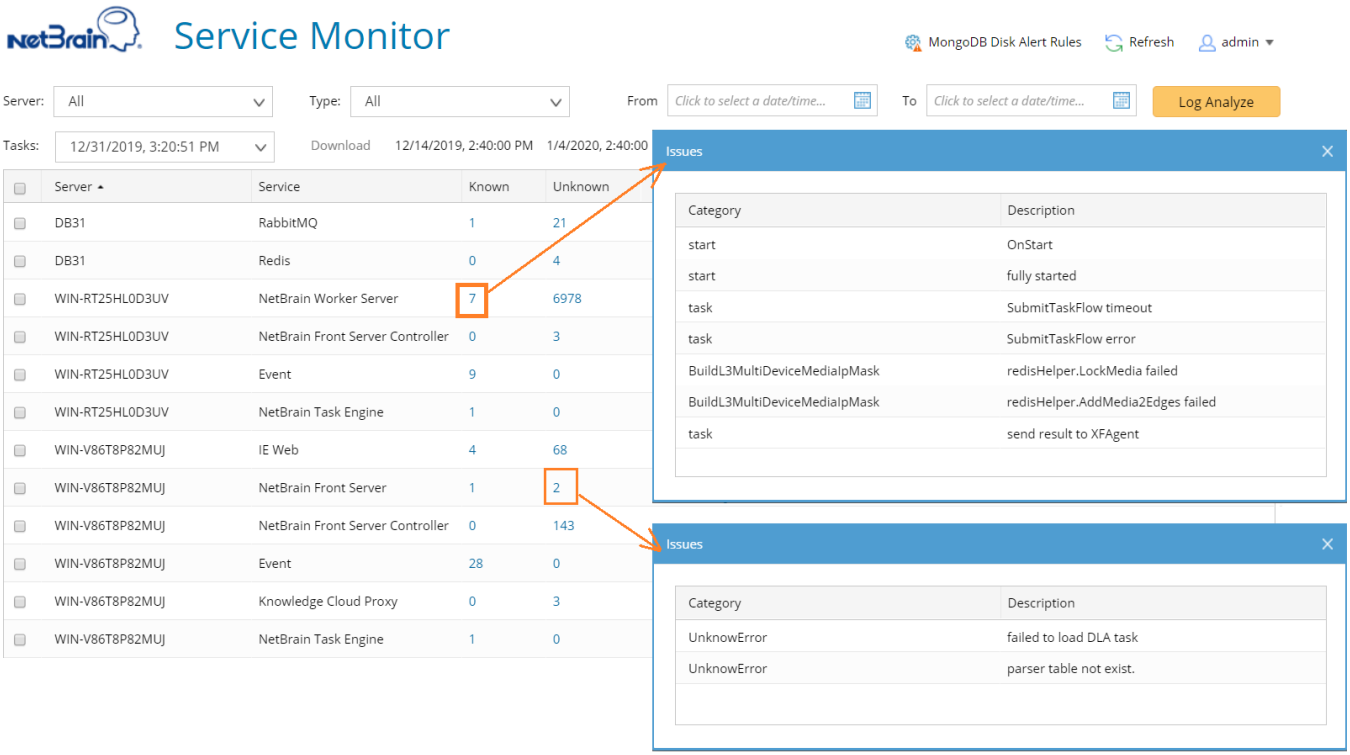
Service anomaly may cause functionality issues in the Thin Client. It usually takes too much time to debug an issue about functions, but finally, it turns out to be caused by stopped services.

IEv8.02 can alert users when any system service has stopped. Users can timely take actions if required, such as restarting these services.



2.10.3. Log Analysis for System Health

In previous versions, the system added the capability to collect Support Logs for troubleshooting purposes. To proactively check system health, IEv8.02 Service Monitor adds the capability to analyze a large number of logs for various NetBrain services and even Windows Event Logs, and extract key words out of them to summarize the addressed issue types and causes. As a benefit, both the Support Team and admin users are able to look into the detailed logs to drill down and resolve the issues before end users might encounter them.



2.11. Enhancements to Change Management

2.11.1. Use Variables in Config-let

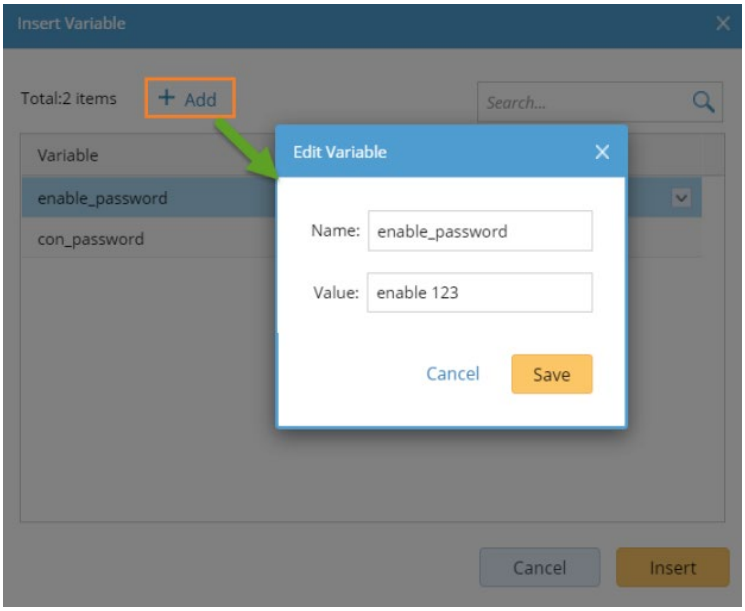
The change configs in Change Management (CM) are displayed in plain text, so there are security risks in some cases. For example, when users modify the device login password in batches through a CM task, it is not secure because passwords are displayed in plain text. Another example is that when users perform ACL security configuration for traffic access control through a CM task, they face the risks of security configuration reveal.

To resolve the security vulnerability and make change config more flexible, IEv8.02 enables users to use variables in config-let. The creator can hide sensitive data with variables when defining a Change node. For example, they can use variables to hide the passwords in a config-let. Only users with specific privileges can view or edit the values of the variables.

Use Flow

1. Add Variables

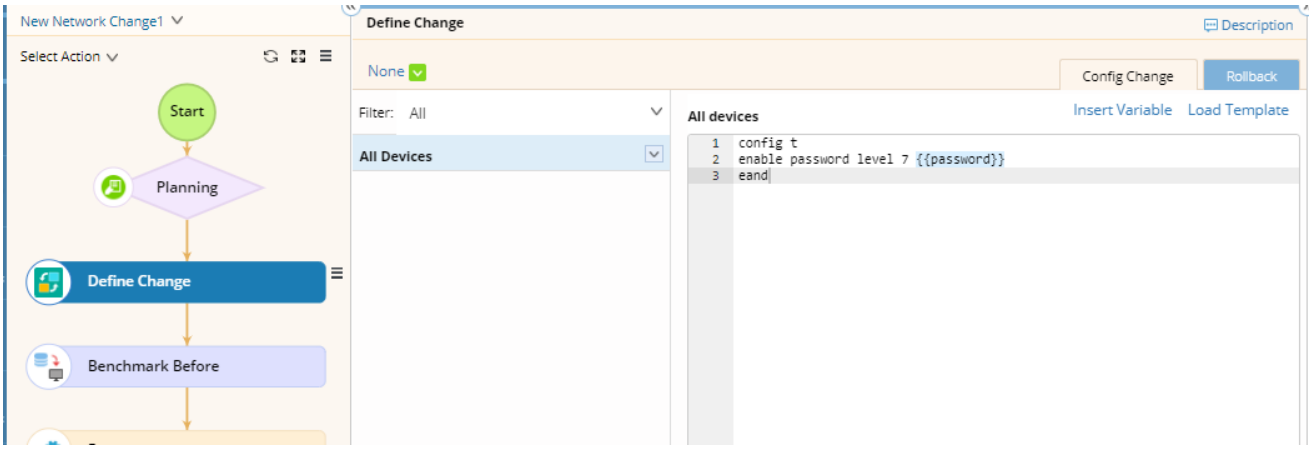
At the **Define Change** node, click **Insert Variable** to add variables and assign values.



2. Replace Data with Variables

Insert the corresponding variable to replace sensitive data you want to protect. The value will be decrypted to form the actual commands sent to the device when the system executes the changes. Enter `{{` and select the corresponding variable.

The system only shows variable names and hide specific data in the commands. The execution logs also hide the specific data.



Only users with specific privileges can edit or view the variables and values.

Privilege	View Variable	Add/Delete/Edit Variable
Create Network Change	Yes	Yes
Approve Network Change	Yes	No
Execute Network Change	Yes	No
View Network Change	No	No

2.11.2. Schedule Change Management Task via REST APIs

IEv8.02 allows users to schedule a network change task via Restful APIs. See section 3.6 for more details.

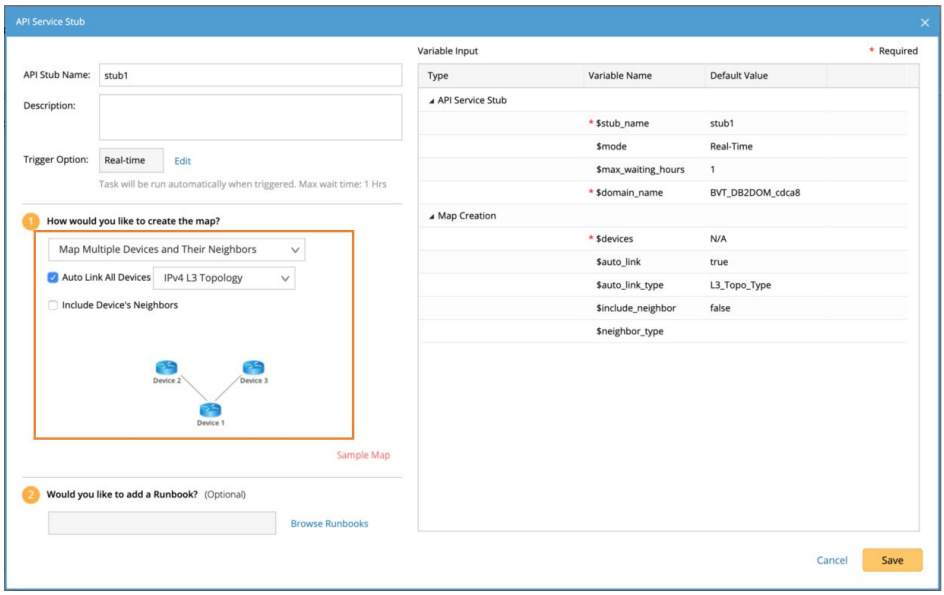
Note: A change management task can only be scheduled to run after being approved.

2.12. More Feature Enhancements

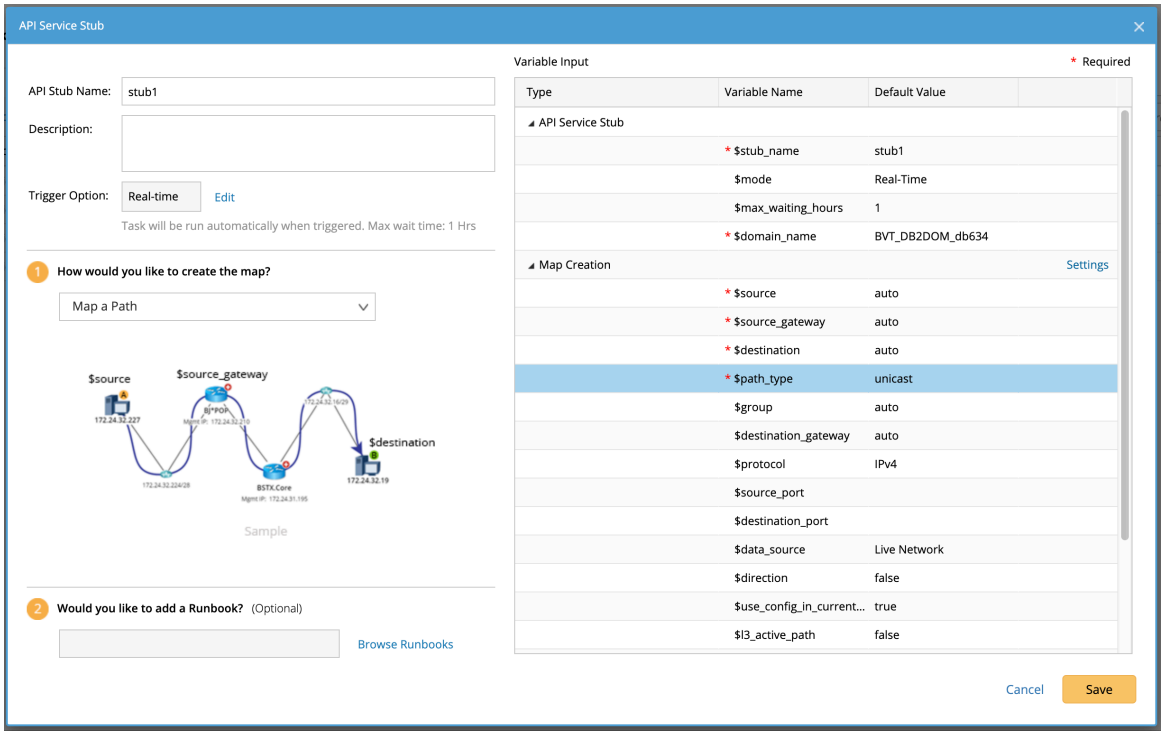
2.12.1. Enhancements to API-Triggered Automation

IEv8.02 has improved the API-triggered task function in the following ways:

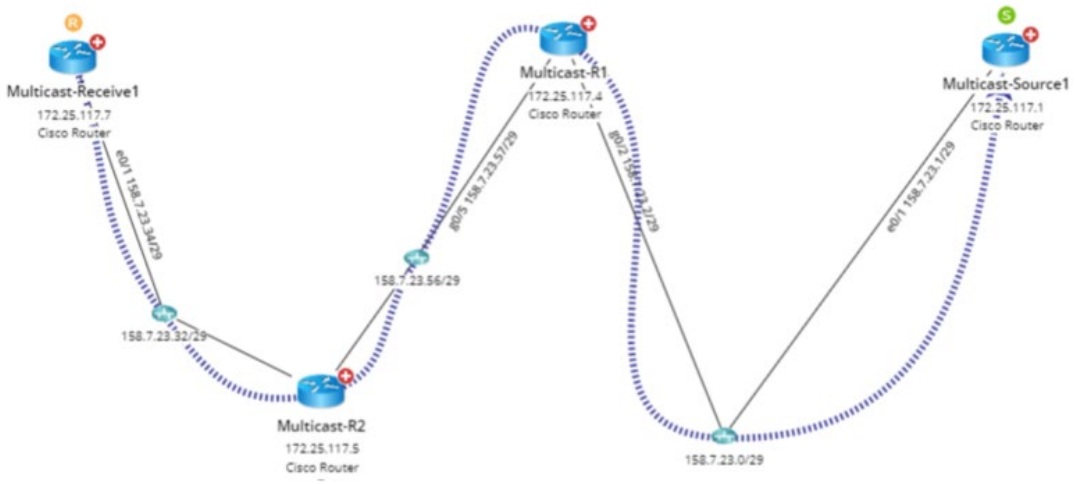
- Allow drawing multiple devices and their neighbors and auto-link them on a map.



- Improve the API-triggered automation to allow searching the basic information of a map under a folder and decide whether to execute the runbook contained in the map.
- Improve the path API to draw multicast reverse paths. Set `$path_type` to multicast, assign the multicast receiver address value to `$destination`, multicast source value to the `$source`, group value to `$group`, and call APIs to draw the multicast path.



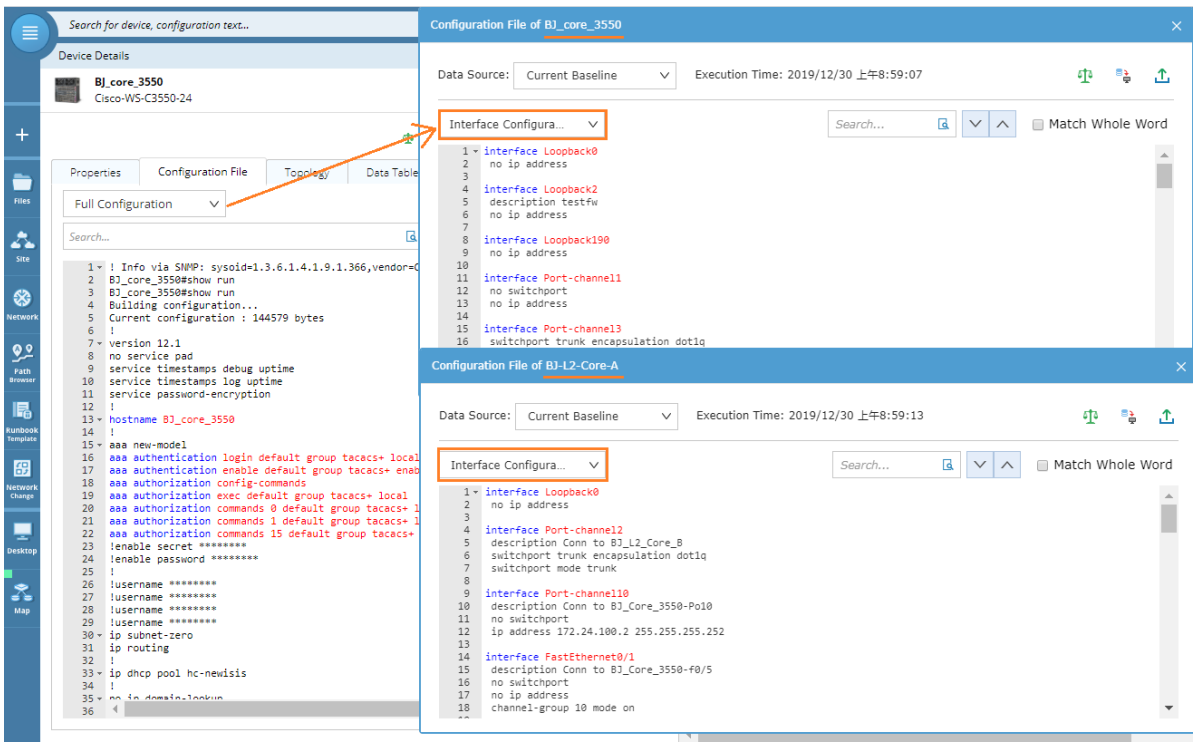
A multicast path is shown as follows:



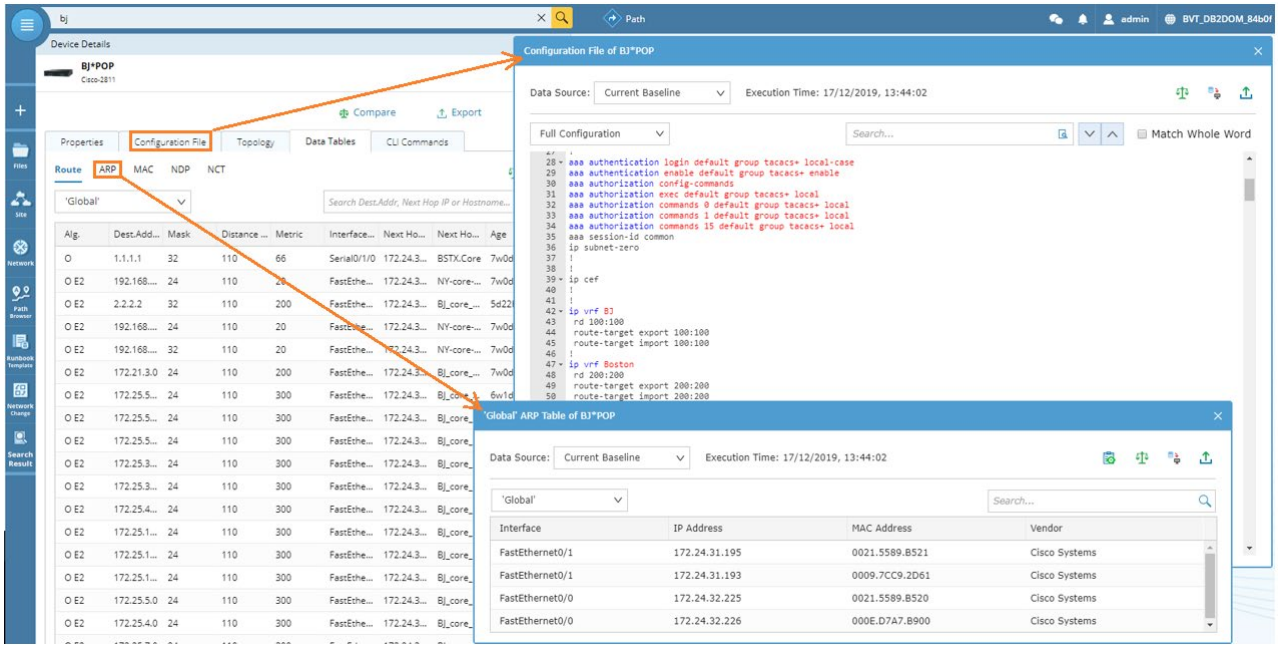
2.12.2. Support the Display of Multiple Device Data Dialogues Concurrently

IEv8.02 enables users to concurrently launch multiple dialogues from the Device Detail pane by introducing non-modal dialogue (resizable). As a benefit, users can view information from multiple device data tables in one screen under the following circumstances:

- Compare the interface configurations between a pair of neighbor devices on a map to troubleshoot the IP mismatch issue.



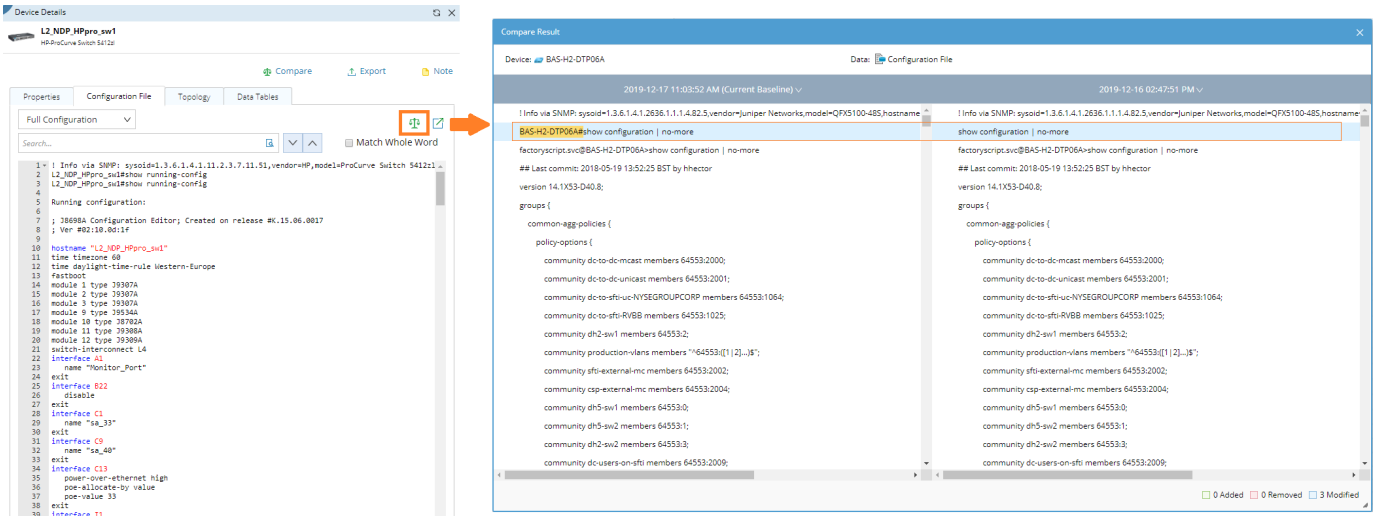
- Check the configuration file, route table and ARP table for a single problematic device.



2.12.3. Enhanced Device Data Comparison Flow

IEv8.02 enables users to directly view the comparison result for device data from the Device Detail pane, including configurations and a variety of data tables.

When users click the Compare button on the Configuration File or Data Table tab, the system directly displays the comparison result rather than launching a Runbook, which requires a few more clicks.



3. Platform Enhancements

3.1. Enhanced Platform Framework

3.1.1.New Multi-CLI Mode Support

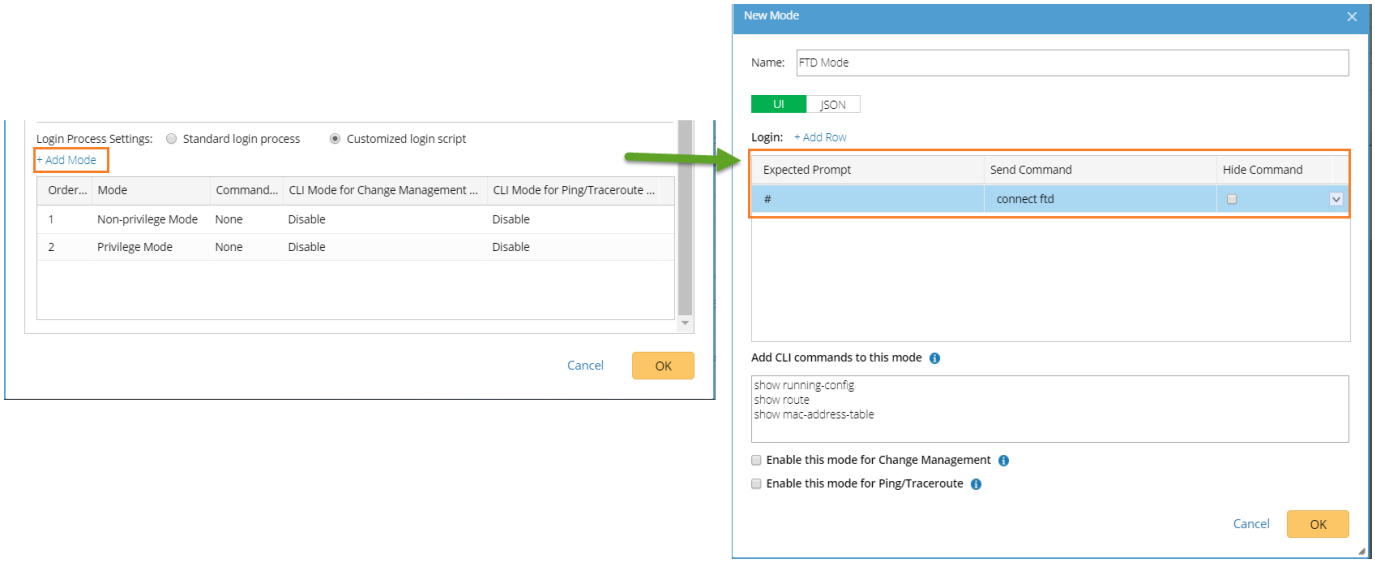
In previous versions, the system allowed users to customize login prompts and commands based on the privilege and non-privilege modes. However, this customization is not applicable when a device type has other CLI modes. For example, a Cisco Firepower includes three modes: FXOS mode, FTD mode, and ASA mode, so the customization based on the privilege and non-privilege modes does not apply to this scenario.

To support more CLI modes and enable users to define commands dedicated to a mode, IEv8.02 optimizes the multi-CLI mode function.

Define More CLI Modes in Driver

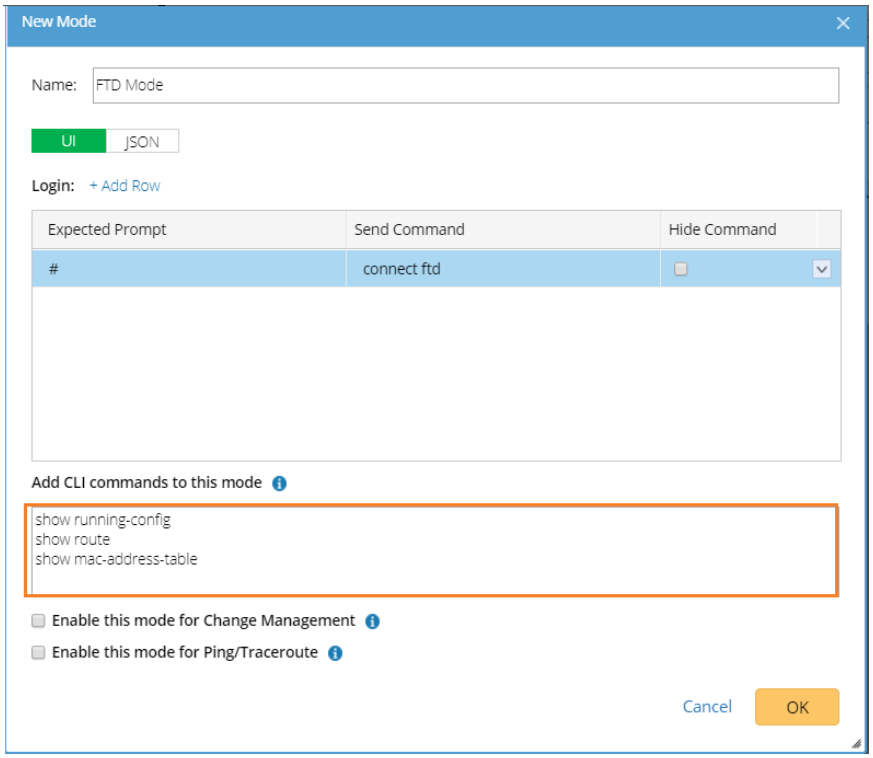
Power users can customize to add multiple CLI modes for a device type in the device driver. The customization includes expected prompts and commands for each mode, and the dedicated CLI commands. During the live access to a device, the system attempts CLI modes from top to bottom until addressing the right one.

Example1: Define a mode and the corresponding prompts



Note: After defining CLI modes in a driver, users can modify the login scripts in the advanced settings of the Shared Device Settings, but the modified scripts will not be synchronized to the driver.

If users specify CLI commands dedicated to a mode, the system will send the commands in the specified mode.



Note: If multiple modes are associated with the same command, the system does not try the command in the subsequent modes when it is executed successfully in a mode.

3.1.2.Build/Retrieve Configuration File for SDN Nodes via TechSpec

In previous versions, the system did not retrieve configuration files for some SD-WAN devices after discovering them via APIs (TechSpec), which caused the problem that Policy and ACL check based on configuration files cannot be implemented in the system.

IEv8.02 improves the configuration retrieval for SDN nodes in the following aspects:

- Enable the configuration file retrieval in TechSpec.
- Add the logic to retrieve configuration files via API > CLI > SNMP when retrieving them in the benchmark/live path/parser/device configuration pane.

3.1.3.New APIs for Plugin/Qapp

IEv8.02 introduces two system APIs for Plugin/Qapp functions:

Category	APIs	Description
Tune	ExcuteSharedTuneDevice() ExcuteTuneOneDevice() ExcutePrivateTuneDevices() ExcutePrivateTuneOneDevice() GetTuneDeviceResultByDevIds() GetPrivateTuneDeviceResultByDevIds()	Tune devices and report tune results.
Hostname Change	GetHostnameChangeList() UpsertHostnameChangeList() KeepLastChangeDevice() SubmitHostNameChangeChildTask()	Detect hostname change and keep the latest hostnames.

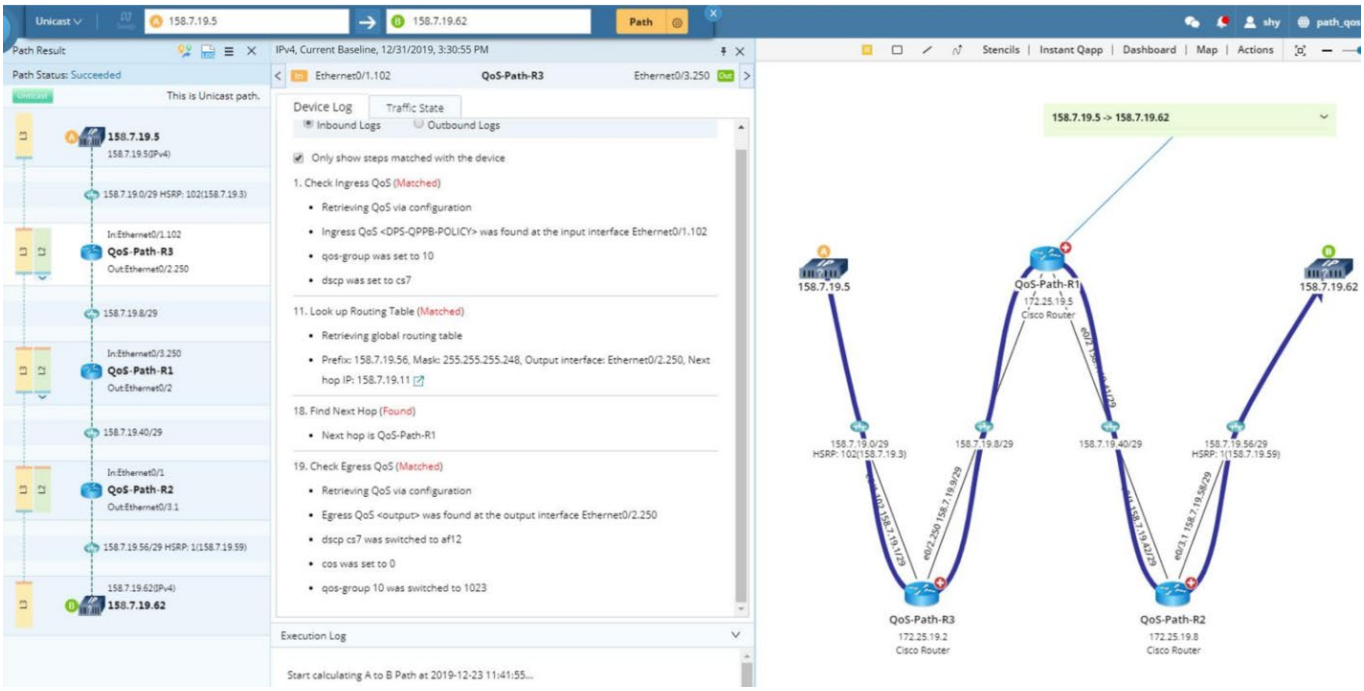
3.2. Extension Based on Enhanced Platform Framework

3.2.1.QoS Path Support

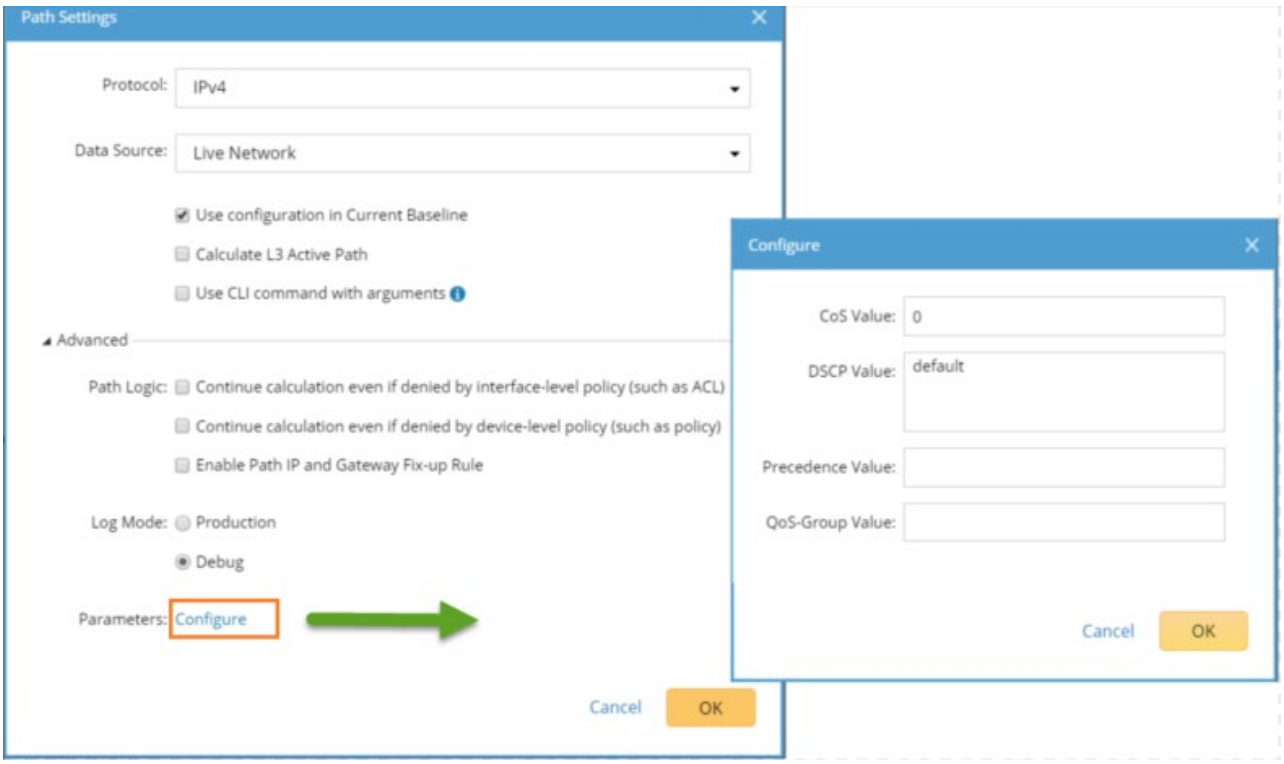
In previous versions, the platform framework provided the capability of traffic path calculation above Layer 4, such as checking DSCP value/URL in ACL or PBR. However, the specific scripts to calculate a QoS path were not available.

IEv8.02 system adds the path scripts to enable the L3 QoS path calculation. The system retrieves QoS parameters such as DSCP, IP Precedence and QoS-Group based on the server-policy information configured

on an interface. It automatically calculates an L3 QoS path based on the QoS policies.



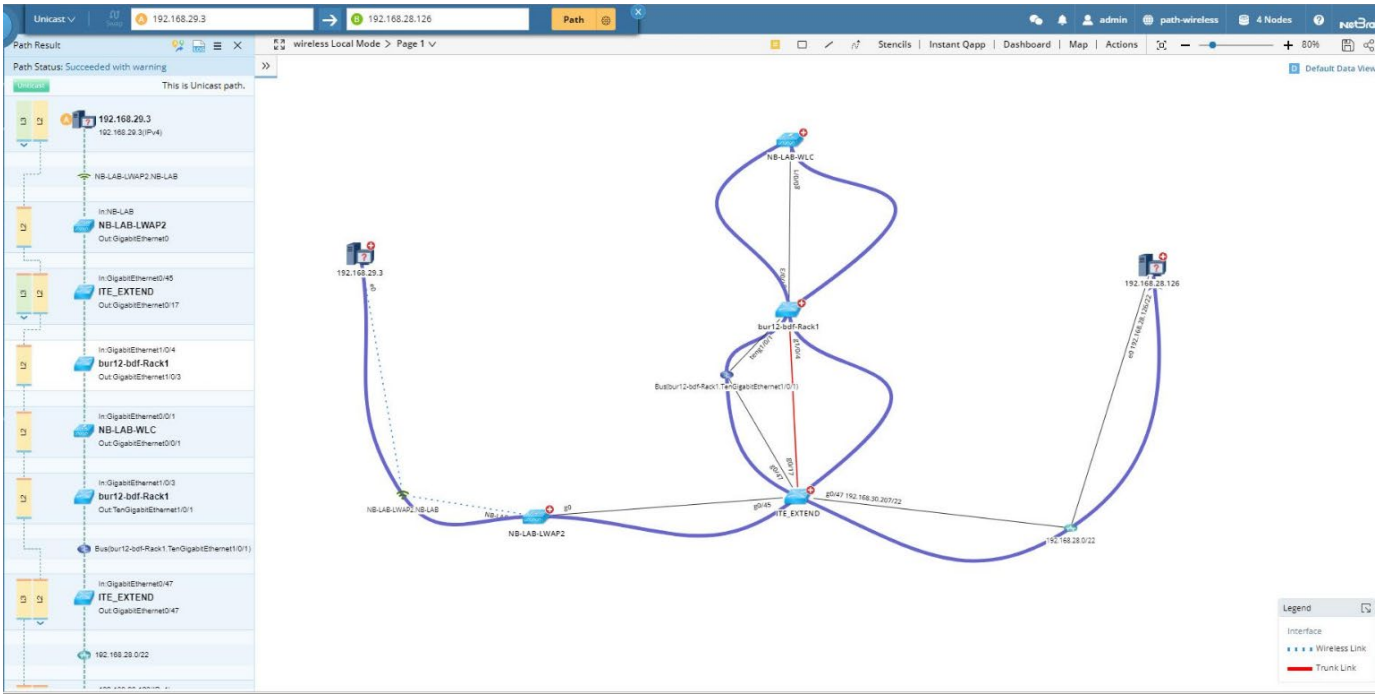
Note: The system cannot recognize and retrieve the QoS information of an end system. When the source device is an end system with QoS configured, end users need to configure the QoS values manually before calculating a QoS path.



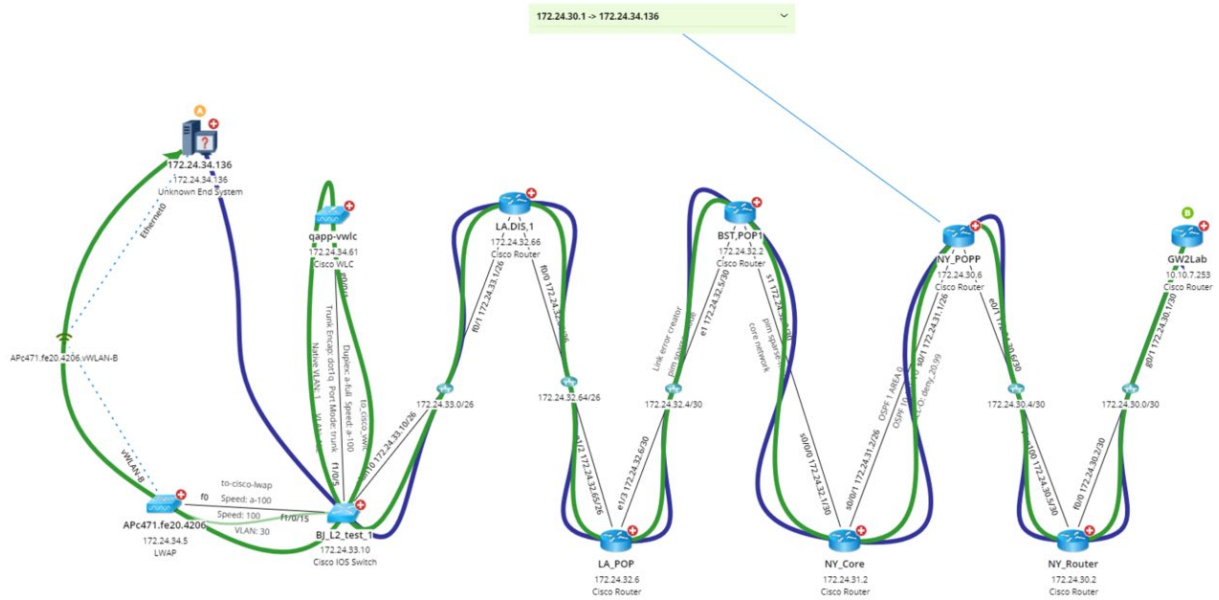
3.2.2.Cisco Wireless L2 Path Support

IEv8.02 improves the support of the L2 path across wireless devices and supports an L2 path calculation across Cisco LWAPs in the Local and Flexconnect mode.

L2 Path across LWAPs in Local Mode:



L2 Path across LWAPs in Flexconnect Mode:



3.2.3.Hostname Change Support to VDC Devices

IEv8.0 introduced the **Hostname Change** function to detect and resolve device hostname duplication issues. It scans device serial numbers and lists the devices with duplicate serial numbers as possible candidates with hostname changes. But this logic does not apply to VDC devices, because each VDC device in a group is a unique device with the same SN. The system always recognizes VDC devices as hostname-changed devices.

IEv8.02 enhances the hostname change logic for VDC devices:

- The system adds a GDR property **VDC_MAC** to differentiate VDC devices in a group.
- The system checks both SN and MAC address during the hostname change check for VDC devices. Only when two devices have the same SN and MAC address, the system determines that one device is the other one's hostname-changed device.

3.3. Path Framework Enhancements

3.3.1.Enhanced Path Debuggability

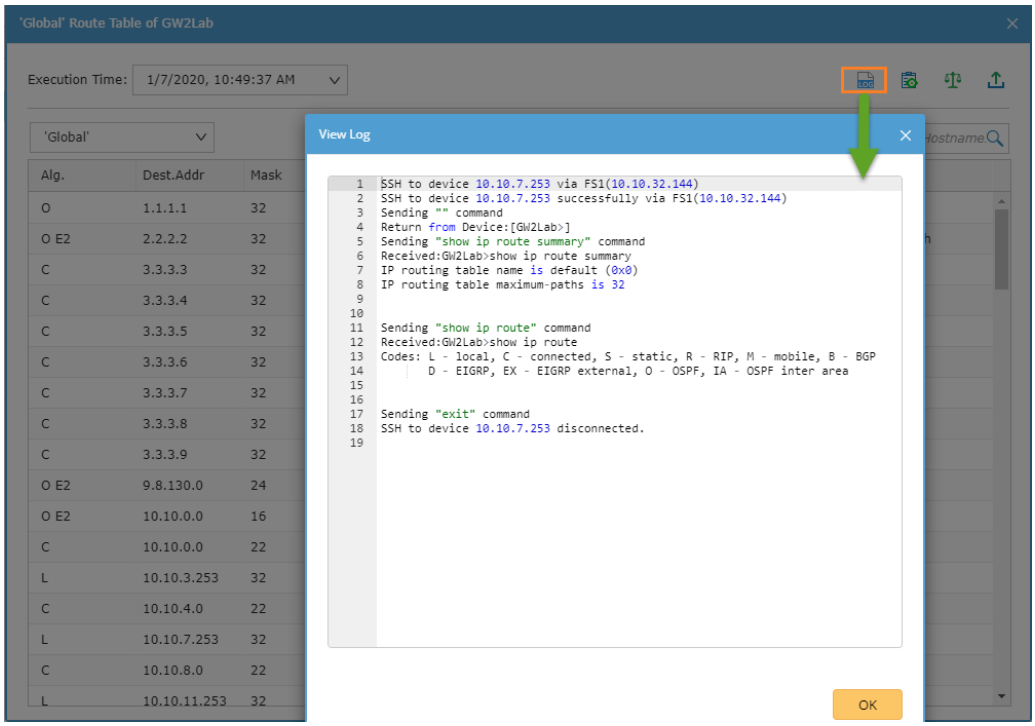
To help end users address the underlying causes of path issues quickly, IEv8.02 introduces the following improvements on path debuggability:

- Add logs for retrieving data tables and CLI commands during path calculation.
- Add path message categories and error tips.

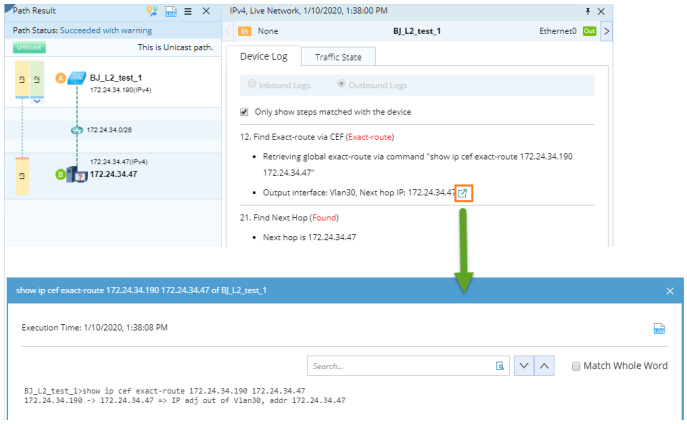
Add Logs for Data Tables and CLI Commands

IEv8.02 records the logs of retrieving data tables and CLI commands during a live path calculation in the Debug mode.

- View the execution logs of system tables and NCTs (network control tables) generated during a path calculation.

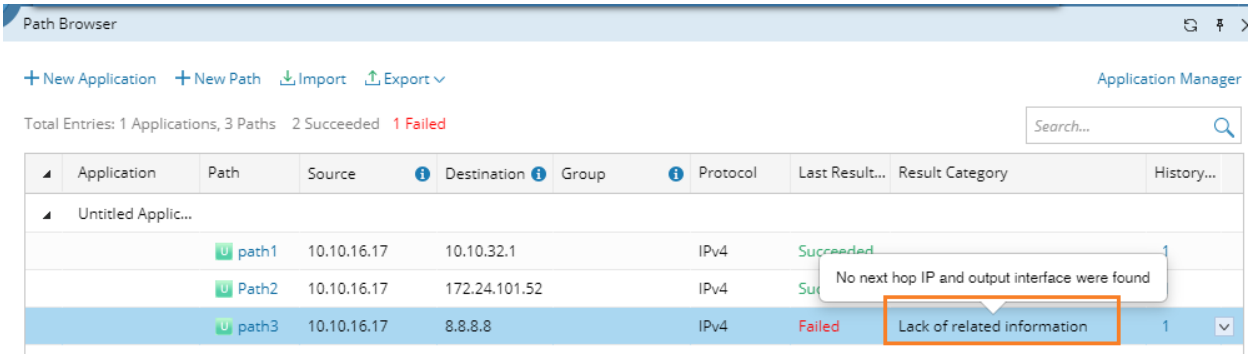


- View the execution logs of CLI commands generated during an L3 active path calculation.



Add Result Categories for Failed Paths

IEv8.02 adds result categories for failed paths in the Path Browser and Application Manager to help identify problem types about failed paths, such as missing data, path logics and framework flaws. The categories enable NetBrain Teams to target the problem scope faster than before and improve the efficiency of resolving path issues.



The information in the result category can be exported along with the path exports in the Application Manager and Path Browser.

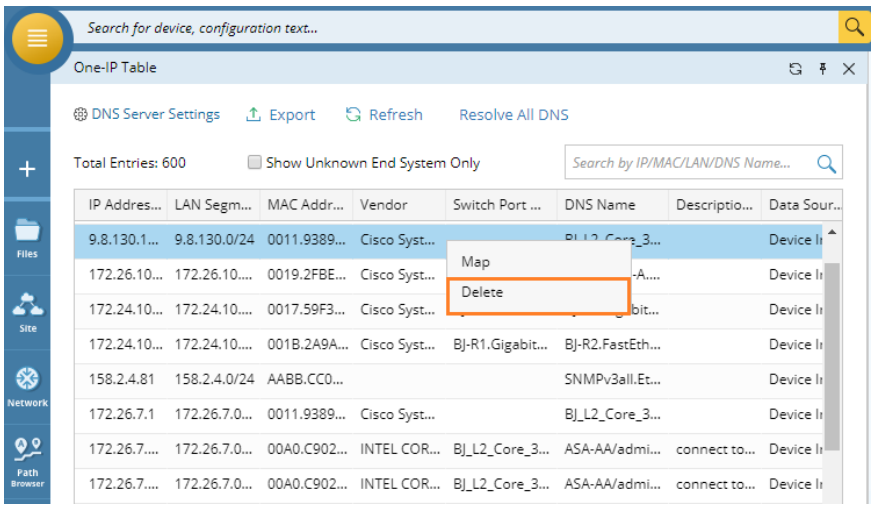
3.3.2.Enhanced Path Calculation for Unknow End Systems

In previous versions, a path related to unknown end systems failed when

- Duplicate unknown end systems were found as a next-hop device during a path calculation.
- Many existing virtual IP addresses were recognized as unknown end systems.

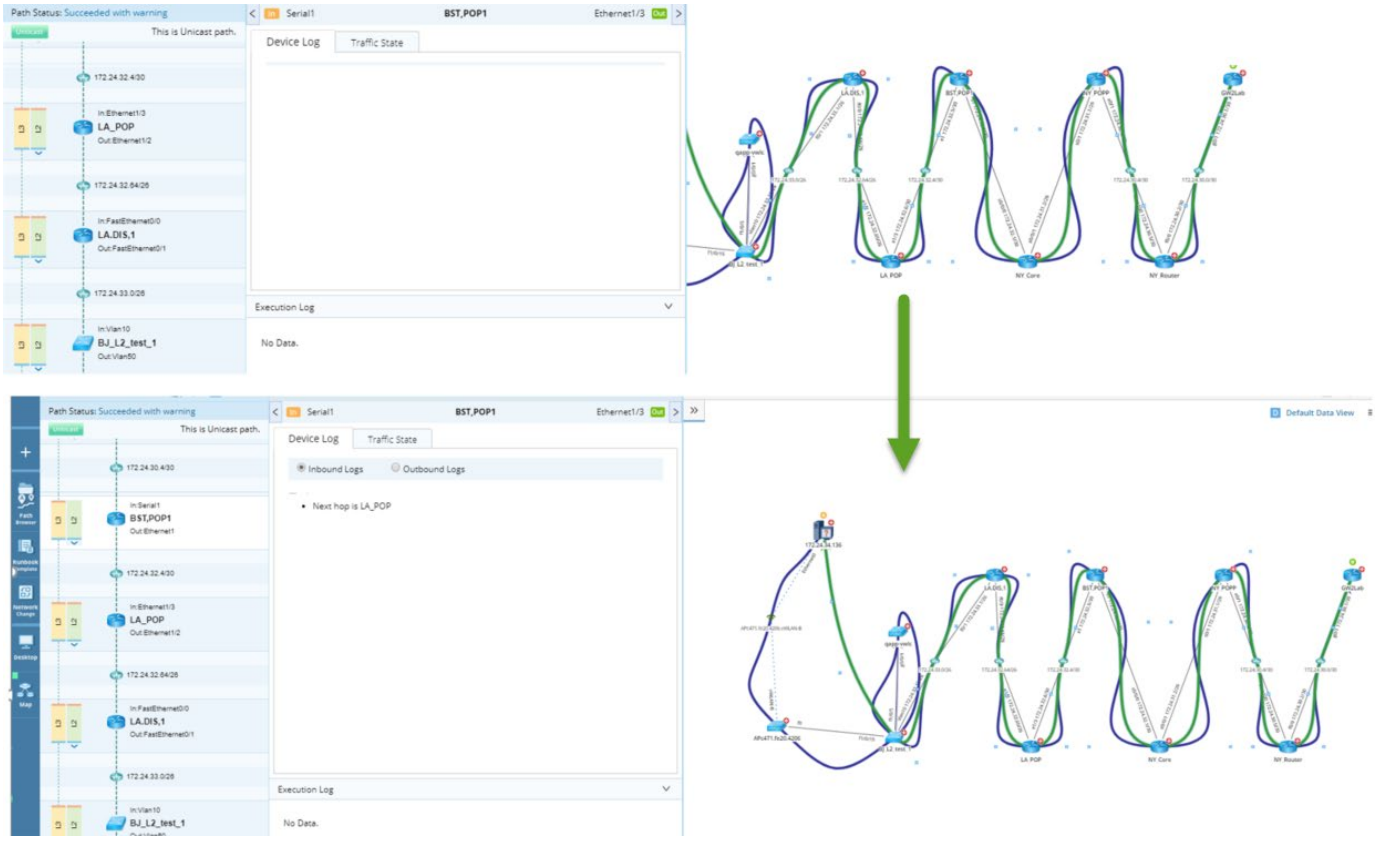
To resolve these issues, IEv8.02 enhances path calculation for unknown end systems:

- Use the latest unknown end system as the next-hop device when the system finds multiple ones during a path calculation.
- Exclude more virtual MAC addresses for Cisco, Juniper, and Fortinet devices, so that the system will not recognize them as unknown end systems.
- Allow users to delete One-IP table entries and unknown end systems together manually.



3.3.3.Pin the Path Result Pane

In the previous versions, some devices might be overlapped by the Path Result pane after calculating a path. To resolve this issue, IEv8.02 adds a Pin function. After users pin the Path Result pane, it will squeeze the map so that the map content can be fully displayed.



3.4. Fully Extensible MPLS Cloud Framework

In the previous versions, MPLS Cloud was used to simulate an MPLS network that carries traffics for customer networks at different regions, which enabled users to calculate the traffics across the MPLS Cloud to make their network complete in the NetBrain system. However, as more cloud deployment scenarios and technologies increase, the limitations of the MPLS Cloud framework arise:

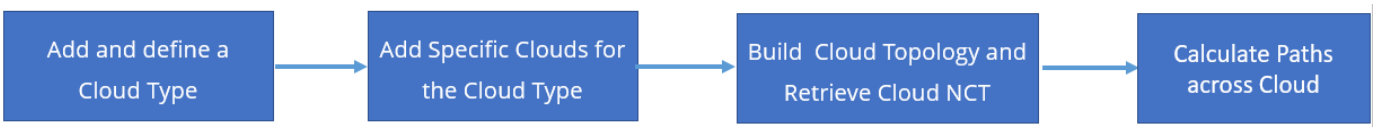
- It cannot simulate PE-side redundancy deployment scenarios because the CE interface cannot be configured to connect to multiple PE IPs.
- It cannot simulate CE-side redundancy deployment scenarios because multiple CE interfaces cannot be configured to connect to the same PE IP.
- It cannot add CE interfaces configured with IP unnumbered configuration.

- It is not extensible to support the deployment scenarios of Carrier Supporting Carriers (CSC) and Inter-Provider.
- It is not extensible to support other cloud technologies, such as VPLS.

To resolve the above limitations, IEv8.02 has redesigned the Cloud framework to make it a generic with high scalability. The new Cloud framework enables NetBrain Platform Team to define a cloud-based via JSON and Python so that they can flexibly adjust and develop cloud functions based on network situations.

Besides MPLS Cloud, the new Cloud framework uses the Cloud concept to represent the network connecting to users' networks but not managed by the users, for different technologies such as VPLS and IPv6.

Use Flow



3.4.1.Add Cloud Type

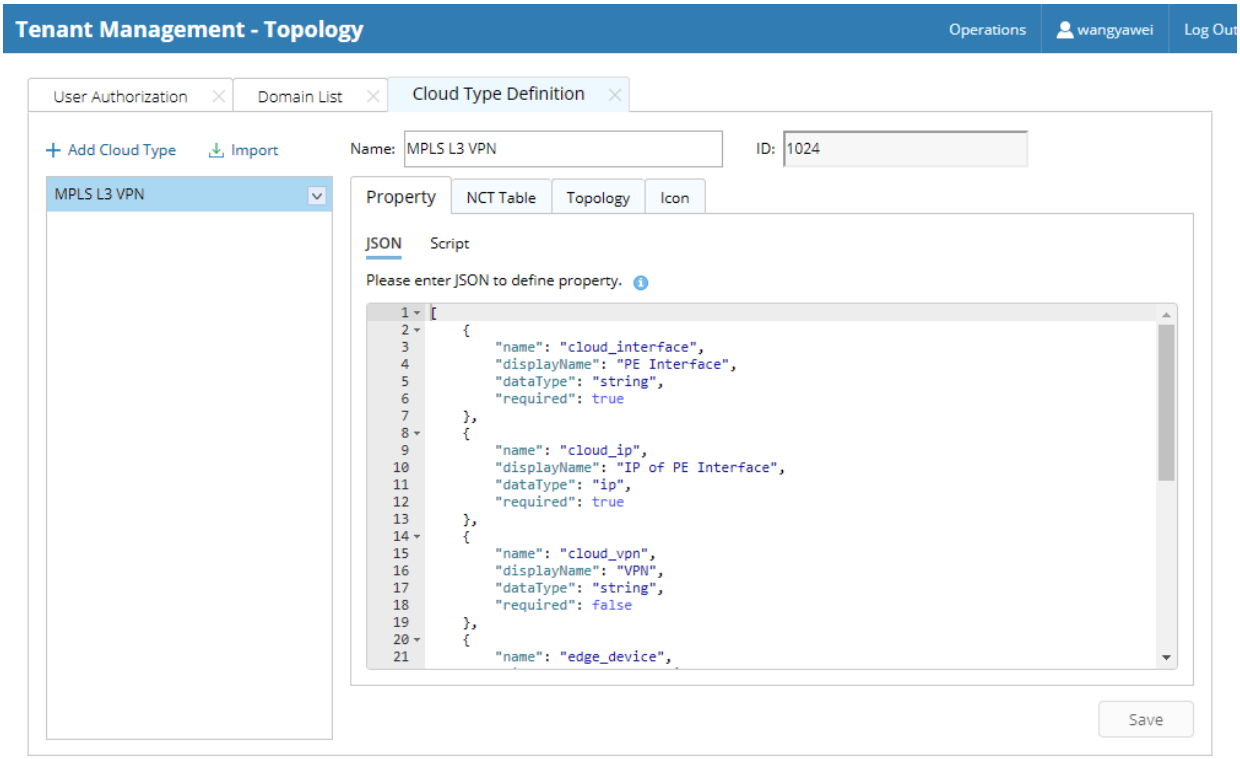
Each cloud type is composed of four components, which are used to build the data model for cloud topology and path calculation:

Component	Description
Property	Define which parameters need to be filled in. For an MPLS Cloud, the parameters include PE devices, CE interfaces, and other parameters.
NCT	Define the NCT needed by a cloud to calculate the path crossing cloud. For example, the virtual route table for an MPLS Cloud, the virtual MAC table for a VPLS cloud.
Topology	Define the logic to calculate the topology for a cloud and its neighbor devices.
Icon	Define the Cloud icon displayed in the system.

Note: After adding or modifying a cloud type, you need to add or adjust the path scripts used to calculate paths for this cloud type.

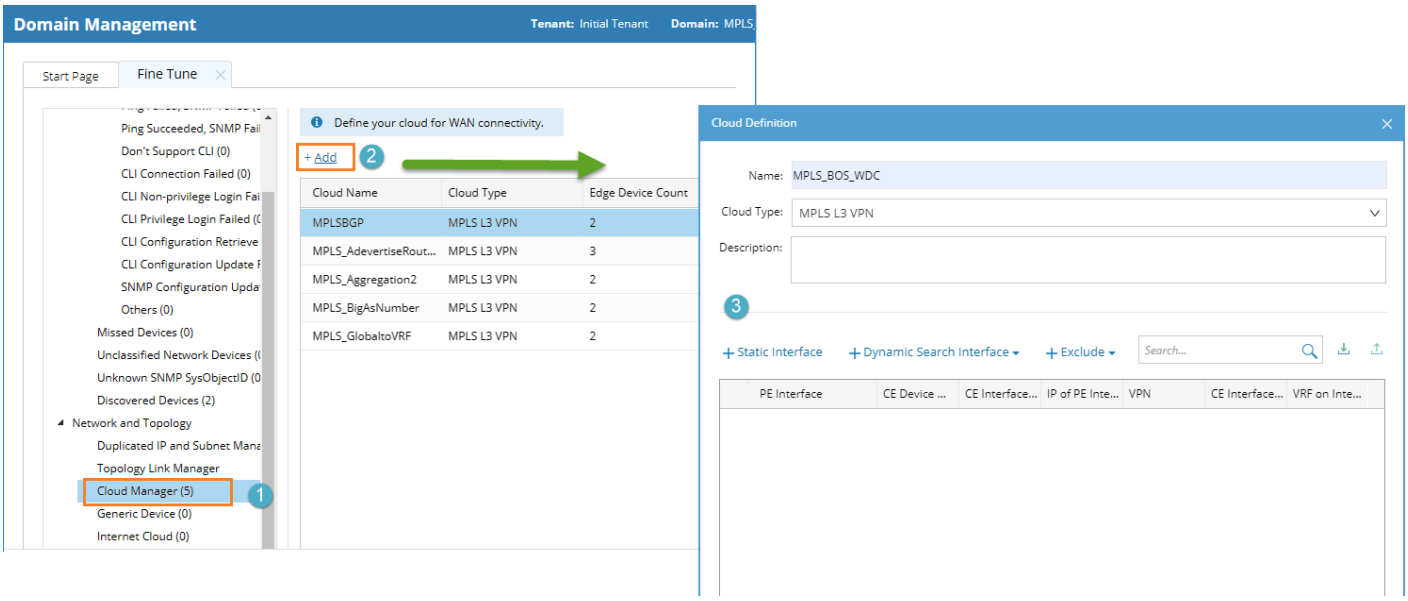
To enable users to use the MPLS Cloud function directly, the system offers a built-in "MPLS L3 VPN" cloud type and power users can define an MPLS cloud as usual based on this cloud type.

Example: Built-in Cloud Type for MPLS L3 VPN



3.4.2.Define Specific Clouds for a Cloud Type

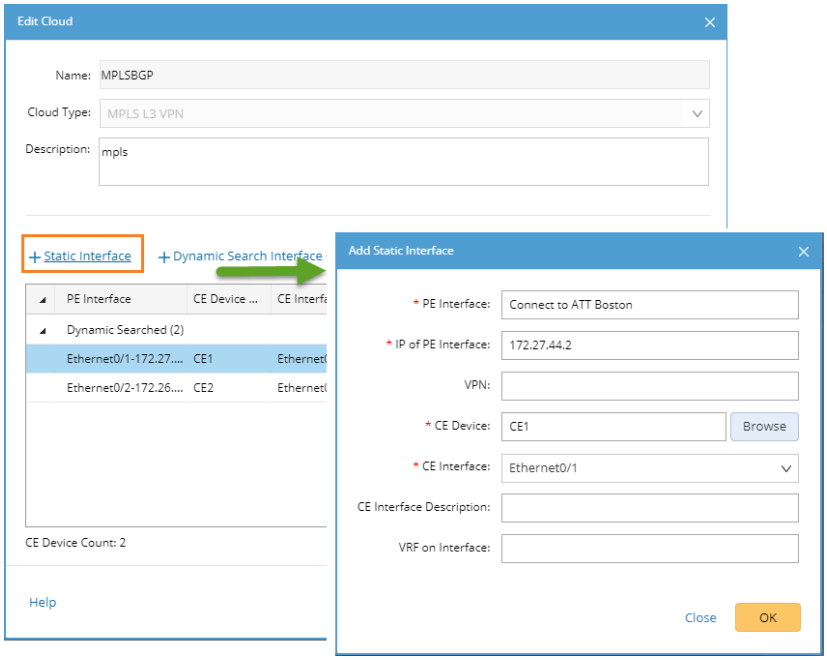
After defining a cloud type, power users can define specific clouds in their domain. After they select a cloud type, the system automatically loads the parameters defined in the cloud type.



Compared with the previous versions, there are some changes in the MPLS Cloud definition:

1. Select **MPLS L3 VPN** as the cloud type.

2. Fill in the PE interface instead of the routing protocol when manually defining an MPLS cloud via Static Interface
3. Fill in any content based on display needs on a map for the configuration of a PE interface. For example, fill in the PE interface “Connect to ATT Boston”, then they will see that MPLS Cloud uses this interface to connect to the CE device on the map.

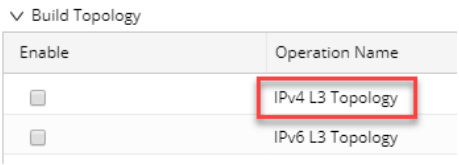


3.4.3.Build Cloud Topology and NCT Data for Path Calculation

After defining a cloud, power users need to build the connection topology between a cloud and its neighbors and retrieve the NCT data used for path calculation through a benchmark task.

Take MPLS L3 VPN for example:

- Build the MPLS topology via the **IPv4 L3 Topology** option.

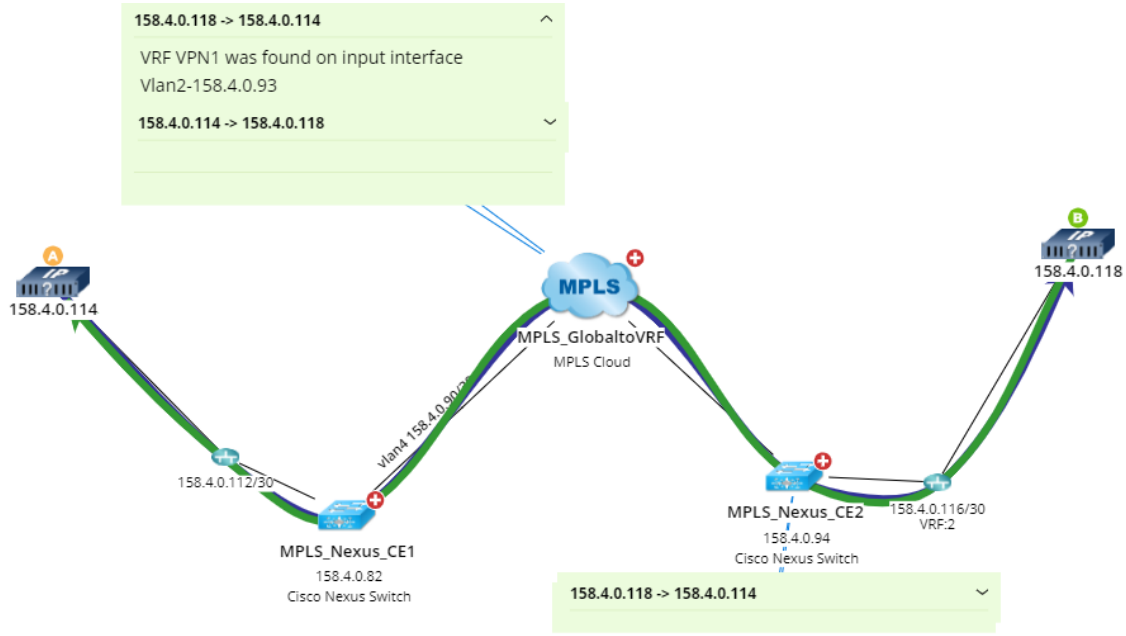


- Calculate the virtual route table via the **Recalculate Cloud NCT Table** option.

3.4.4.Calculate Path Across a Cloud

After building the topology and NCT data, end users can calculate a path across the defined cloud.

Example: A Path across MPLS Cloud



3.5. L2 Topology Improvements

3.5.1.Auto Clean Unknown End Systems

For a network using DHCP, an IP address may be assigned to different PCs at different times. This causes multiple unknown end systems with the same IP address, but different MAC addresses existed in the domain. When the system calculates a path across this kind of unknown end system, the path will fail due to duplicated MAC entries.

To resolve this issue, IEv8.02 will automatically clean outdated unknown end systems when a One-IP table entry is deleted via either of the following methods

- When a One-IP table entry older than the specified days (auto clean) is automatically deleted from the system, the unknown end systems generated by the One-IP table entry will be deleted.

Data Type	Data Size	Auto - Clean Rule
Qapp/Gapp Execution Logs	N/A	☑ Delete data older than 14 days
One-IP Table Entries	4KB	☐ Delete data older than 14 days

- When One-IP table entries are deleted at L2 topology rebuilding after enabling the DHCP Filter function, the unknown end systems generated by the One-IP table entries will be deleted.

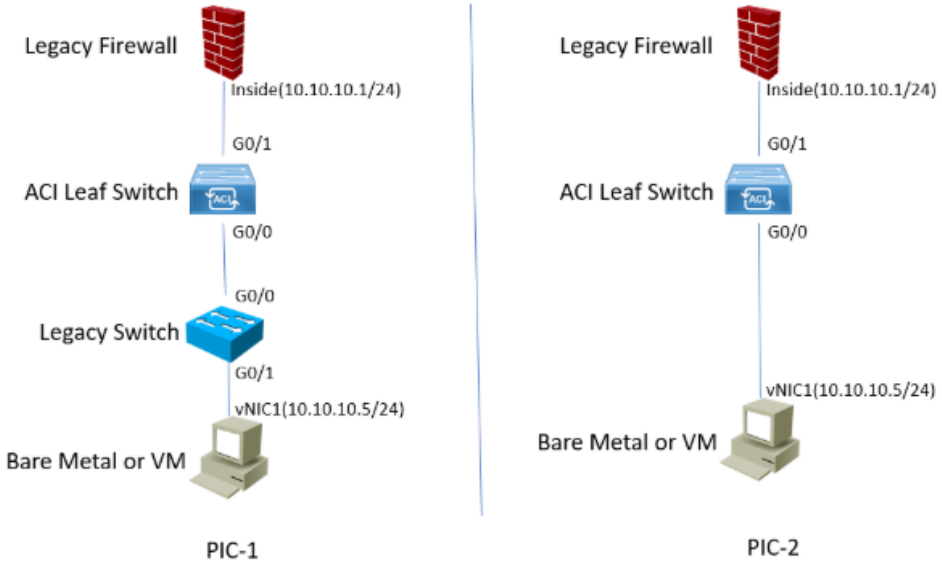
Tip: When the DHCP filter is enabled, the system filters out the outdated DHCP entries and only keeps the latest IP addresses when building build L2 topology.

3.5.2.Enhanced L2 Topology of SDN Nodes

IEv8.02 has enhanced the L2 topology algorithm to improve the accuracy of L2 Topology in the virtualization, legacy and SDN mixed network.

Gateway of a VM host in Legacy Network

In the previous versions, when the gateway of a VM host was a legacy device and the ACI Leaf Switch connected to the gateway worked as a pure L2 switch, the system cannot calculate the L2 path in this network scenario because of topology issues.

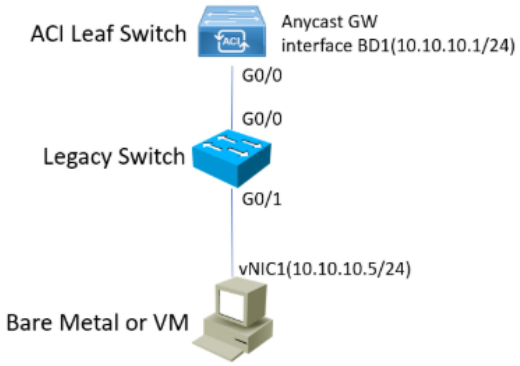


IEv8.02 has resolved the L2 topology connection issues and supported the path calculation by optimizing topology calculation logic: When generating an end system, if an entry in the Global Endpoint Table has only MAC Address without IP Address, the system uses the MAC Address to find the IP + MAC entry without a

switchport in the One-IP Table, generates a legacy end system and then save the L2 Topology relationship to ACI L2 Topology Table.

Gateway in ACI and End Point Connecting to legacy Switch

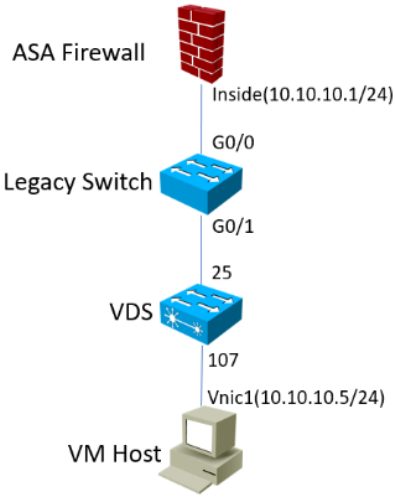
In the previous versions, when the gateway of a VM host was an ACI Leaf Switch and there was a legacy device between the gateway device and VM host, the system cannot calculate the connection between the VM host and gateway device.



IEv8.02 has fixed this issue by optimizing the L2 topology calculation logic in this scenario.

VM Host's L3 Neighbor is ASA Firewall

An ASA Firewall does not support NDP. In the previous versions, when a VM host connected to VDS → Legacy Device → ASA Firewall (shown as the figure below), the system would generate two L2 topology connections for the VM host, one calculated based on legacy topology logic (VM Host Vnic 1 ←→ Legacy Switch G0/1), the other based on SDN L2 topology logic (VM Host Vnic 1 ←→ VDS107).



IEv8.02 adds the legacy topology to Multi-Source Topology Priority Rules and relies on the following priority to select topology when a topology is generated from different sources: NSX > vCenter > Legacy > ACI.

Duplicate VRF and IP Subnet

In the previous versions, when ACI Fabric Instances deployed in different regions were configured with the same VRF and Any Gateway, these ACI instances would connect to the same one LAN Media when NetBrain calculated topology for them.

IEv8.02 has resolved the issue by generating and assigning different zones for ACI Fabric instances.

Virtual Network Device installed on VM Host via Image

In the previous versions, when a virtual network device (such as a virtual ASA) was generated on a VM host via installing network device image, the system treated the virtual network device and host as two devices, and L2 topology generated in this system were not incorrect in this scenario.

IEv8.02 has improved the topology logic to discover them as one device.

3.6. New REST APIs

IEv8.02 introduces the following new REST APIs:

API	Description
Get License Node Info	Call this API to get the overall system license node information.
Get Production Info	Call this API to get the information related to the product version.
Schedule Change Management Task	Call this API to create a scheduled task for an approved network change.
Get a Scheduled CM Task	Call this API to get a scheduled task of an approved network change.
Update a Scheduled CM Task	Call this API to update a scheduled task of an approved network change.
Delete a Scheduled CM Task	Call this API to delete a scheduled task in the “Waiting” status.

4. Security Enhancements

4.1. Limit Guest User's Privilege

By default, the system only grants Guest Users with limited access to system resources by disabling their privilege of Shared Resource Management. In previous versions, the privilege of Shared Resource Management only covered most resource types in the system, such as Qapps, Gapps, Parsers, Runbook Templates, etc., but not all of them. It might cause potential security risk that Guest Users can modify or delete public resources, such as public maps and A-B paths.

To limit Guest User's privileges, IEv8.02 expands the privileged scope for Shared Resource Management to cover more resources, including:

- One-IP Table
- Inventory Report
- Device Property
- CLI Commands Template
- My Files/Desktop
- Path Browser Pane
- Network Pane
- Application Manager
- Static Data View Manager
- Context Action Manager
- System Automation Task Manager

4.2. Upgrade Third-Party Components to the Latest Version

To ensure the longevity of support and the most up-to-date code from a security standpoint, IEv8.02 upgrades the following components to their latest versions.

Component	From Version ..	To Version ..
RabbitMQ	3.7.14	3.8.1

Component	From Version ..	To Version ..
Python	3.7.1	3.7.5
Gojs	1.6.9	2.0.18
Node.js	8.2.1	9.3.0
OpenSSL (Windows + Linux)	1.0.2p	1.0.2t
psutil	5.5.1	5.6.7

4.3. Validate Data Access Privilege for Extended Device Scope

The device access policy introduced in the previous version allows specified users to have the privilege of data access to specified devices. However, it might be out of control when it comes to Qapp execution because the device scope of Qapp might be expanded to neighbor devices depending on the device input settings for a Qapp.

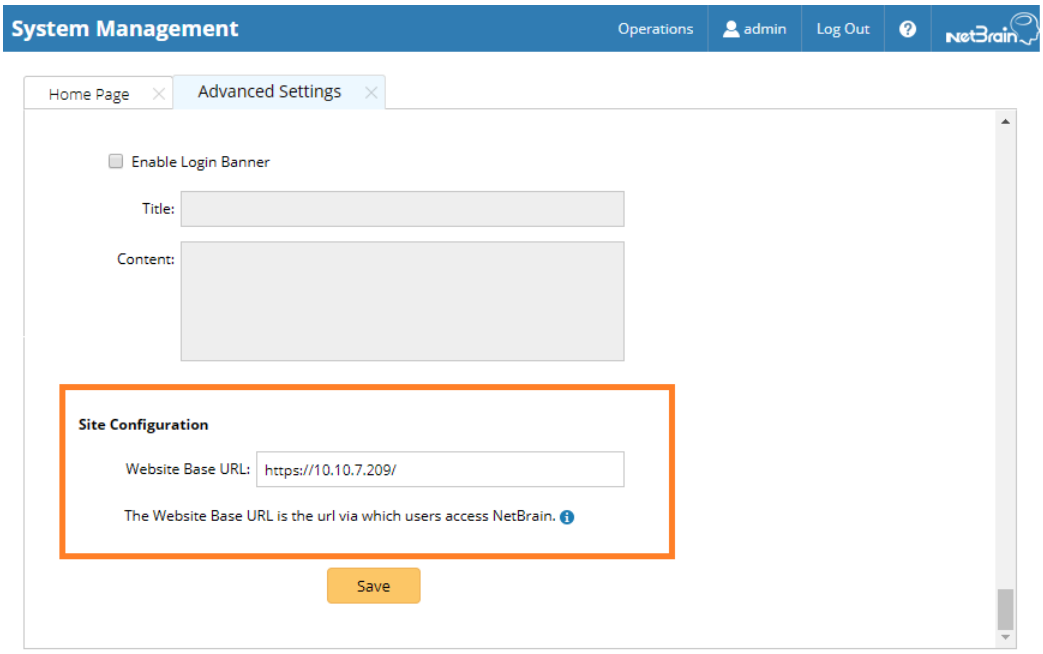
To fix this vulnerability, IEv8.02 starts to validate the data access privilege on the extended device scope for Qapp execution. If the executor is not authorized to view the device data for any extended neighbor device, the system will not run Qapp on that device.

4.4. Single Source for Website Base URL

In previous versions, when users wanted to share a map or reset a password, the system provided an assembled URL for redirection by directly obtaining the domain URL from the client's web browser cache, which may have potential security risks.

To fix this vulnerability, IEv8.02 enforces the definition for a base URL and uses it as a single source for website URL assembly. Every time a user requests a website URL, the system uses the predefined base URL to assemble it, rather than obtaining the base URL from the user's web browser cache.

IEv8.02 guides the system administrator upon login to configure the base URL at **System Management > Advanced Settings > Website Base URL**.



4.5. Other Enhancements

- Clear Browser Temporary Data after User Logout

  Note: Refreshing webpage will not clear browser temp data.
- Disable Insecure Communication Protocols (SSL and TLS < 1.2) Unless Required
- Protect Redis Sentinel by Authentication and Rename Default Command
- Remedy Aged Libraries for Redis
- Fix REST API Vulnerability about Roles and Privileges

5. Performance Enhancements

Compared to previous versions, the system performance to load the site tree and the device group tree with a large number of entries has been improved a lot in many features.

Category	Test Scenario	IEv8.01	IEv8.02
Load Site Tree	Load a tree with 9000 sites (5 devices per site) in: ▪ Site Panel ▪ Site Manager ▪ Dynamic Search ▪ Select Device/Interface ▪ Inventory Report	3~4 minutes or failed	1~3 seconds
Load Device Group Tree	Load a tree with 3000 device groups (50 devices per group) in: ▪ Device Group Panel ▪ Dynamic Search ▪ Select Device/Interface ▪ Network Panel ▪ Plugin Manager ▪ Search Result ▪ Add to Device Group ▪ Scheduled Task ▪ Tune Note: This improvement is not significant in **Shared Device Settings > Apply to Device Group**.	5~20 seconds or failed	1 second

6. Known Issues

The known issue (ENG-69067) for **upgrading from IEv8.0 or IE8.01 to IEv8.02** is as follows:

- If the system upgrades without any of components: License Agent/Elasticsearch/RabbitMQ/Redis, user's input for SSL files (Public Key, Private Key and CA) is required no matter what the component is lack, and there is no validation for the mismatching between the newly uploaded certificates and previous ones.
- Changing SSL files (Public Key, Private Key and CA) is not allowed.