



NetBrain[®] Release 11
Release Note

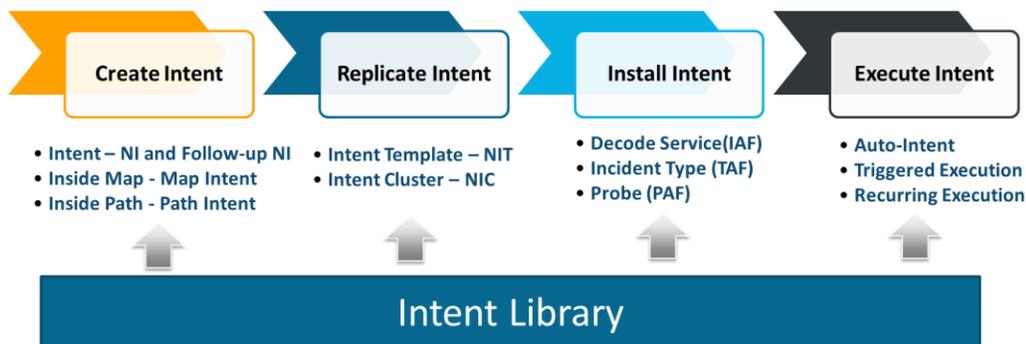
Table of Contents

Table of Contents	2
1 Key New Features and Enhancements	4
2 Create Intent	7
2.1 Copy Verified Parser	7
2.2 Enhance Diagnosis Definition	8
2.2.1 Support Inline Expression.....	9
2.2.2 Support NIC as the follow-up NI.....	10
2.2.3 Support Elseif branch in diagnosis	11
2.2.4 Better support for the table-based diagnosis	12
2.3 Intent Manager.....	17
2.4 Visual Parser Enhancements	18
3 Replicate Intent	20
3.1 NIC Improvements.....	20
3.1.1 Auto Mode	20
3.1.2 Auto Test.....	22
3.2 Intent Template (NIT).....	23
3.2.1 Define NIT	23
3.2.2 Install and Use NIT	24
4 Install Intent	25
4.1 Intent Library	25
4.2 Auto Intent	29
4.3 Triggered Automation (TAF).....	31
4.4 Preventive Automation (PAF).....	34
5 Use and Execute Intent.....	36
5.1 Clone and Execute NI from Map (IAF)	36
5.1.1 Published Intents	37

5.1.2	Map Intent	38
5.2	Path Intent.....	40
5.2.1	Create a Path Intent Via Auto Intent.....	41
5.2.2	Create a Path Intent by Cloning from NIC/NIT.....	42
5.2.3	Run Path NI and View Results	42
5.3	Diagnosis Tree	43
5.4	Self-Service Tools.....	46
5.4.1	Incident Pane/Portal Enhancements	47
5.4.2	Teams Chatbot	49
6	Other Enhancements.....	51
6.1	Incident-based Collaboration Workflow Enhancement	51
6.2	Benchmark Schedule Enhancement.....	52
6.3	Email Change Analysis Report	54

1 Key New Features and Enhancements

NetBrain R11 greatly enhances the Intent-based **Problem Diagnosis Automation System (PDAs)**, which automates the Diagnosis of repetitive problems and enforces preventive measures across the entire network. Especially, R11 improves the flow of building a PDAS system on a large scale with the following new features and enhancements:



- [Create Intent](#)

R11 continuously improves the NI editor: allows using the verified parsers, supports NIC as the follow-up diagnosis, adds the inline expression, provides a better table-based diagnosis, etc. Also, a standalone Intent Manager is added to manage the common intents, path intents and newly added map intents.

- [NIC Enhancements](#)

R11 adds two new features of the Intent Cluster (NIC), auto mode and auto test. The Auto Mode reduces NIC definition from 7 steps to three simple steps, and the auto test simplifies the definition of the device classifier and eigen grouping.

- [Intent Template \(NIT\)](#) ^{New}

R11 introduces NIT, a breakthrough that greatly simplifies the intent's replication. Users can define NIT for any NI with the single-device diagnosis by setting the device qualifications and the critical variables to be auto-tested. The intent with the template configured can be cloned on the fly after it is decoded and installed in the Intent Library to be executed interactively (IAF), Triggered Automation (TAF), and Preventive Automation (PAF).

- [Intent Library](#)^{New}

The intent library is the central console to install Intent and NIT in the PDAS system: enable the backend decoding service to qualify the devices against the template setting; update the baseline data; enable the auto intent to create the intents in the map (Auto Intent for the IAF); install the intent in TAF and PAF; execute the configure orchestration files downloaded from NetBrain KC to download and install the intents automatically.

- [Interactive Automation \(IAF\)](#)

R11 greatly improves the Intent pane of the map with three new features: [Auto Intent](#)^{New}, creating the intents from the qualified NIT for the map devices on the fly; [Published Intent](#)^{New}, displaying the published intents for the map devices; [Map Intent](#)^{New}, a dedicated NI for the map.

- [Triggered Automation \(TAF\)](#)

R11 introduces a new Incident Type, Virtual Incident Type (VIT), to encapsulate the detailed logic of Hash Tag and remove the hurdle of the diagnosis by the Hash Tag, which can be associated with the incoming Incident Type to match intents by the Hash Tag. The self-service applications can use VIT directly.

- [Preventive Automation \(PAF\)](#)

The intent timer is added to the PAF, and the intent template can be installed in PAF.

- [Execute and Use Intent](#)

R11 greatly improves the Diagnosis Tree: add device nodes between the diagnosis node and NI node and add the detail pane for each node, such as the device and diagnosis pane. The intent results can also be displayed as a special data view on a map.

- [Self-Service Tools](#)

Besides improving the Incident Pane/portal, R11 creates a new self-service, [Teams Chatbot](#)^{New}, as a simple way to run NetBrain automation functions and view the results in teams.

- [Incident-based Collaboration Workflow](#)

R11 Improves the search IntelliSense pane to show more related results and allows users to create the map directly from an entry. A new Incident entry is added to the map.

2 Create Intent

The first step in building the PDAS system is to define the base NI. R11 makes many improvements to ease configuring diagnosis and better support the table formatted data.

2.1 Copy Verified Parser

In the first step of **Configuration Diagnosis, Define Baseline**, R11 support copying the Verified Parser besides the library. The verified parsers are from the results of the Intent Template (NIT) decode engine. The system already verifies the historical commands and the corresponding parser.

The screenshot displays the Configuration Diagnosis interface. The main window is titled "Configuration Diagnosis" and shows a progress bar with three steps: "1. Define Baseline", "2. Define Diagnosis", and "3. Debug". The "1. Define Baseline" step is active. The "Baseline Data" panel on the left shows a sample configuration for a Cisco device. The "Define Variables" dialog box is open on the right, showing a list of "Verified Parser" options. The "Copy Parser" button is visible. The list of parsers includes:

- conf-ACL-extended(Parser1)
- path- (NIC)(Parser1)
- design- (NIC)(Parser1)
- shell-intent(Parser1)
- 3 - L3 OSPF Check - Cisco IOS(Pars...
- 1-Weekly-Config-Analysis(Parser1)
- WAN-LINK-INFO(Parser1)
- 3-3 ASSE - Device Access Security ...
- conf-prefix(Parser1)
- conf-ACL-standard(Parser1)

The "Output: Table Variable" is shown for the selected parser. The "Add Text" option is also visible, with the description "Use one or a few lines of input text as Parser. Output: Multi-Line String".

2.2 Enhance Diagnosis Definition

In the second step of **Configuration Diagnosis, Define Diagnosis**, R11 provides many enhancements such as:

1. Support the inline expression for the variables.
2. Support the sub table creation (when the option **loop table rows** is enabled) and the whole table operations (when the option **loop table rows** is disabled).
3. Support break the current loop of a table.
4. Support Elseif branch.
5. Support NIC as the follow-up intent.

The screenshot displays the 'CLI Command Diagnosis' window for device 'US-BOS-R1' with the command 'show interface'. The interface is divided into three steps: '1. Define Baseline', '2. Define Diagnosis' (active), and '3. Debug'. The left pane shows a list of interface status commands and their original text. The right pane is the configuration area for 'Untitled Diagnosis 1'. It includes a 'Type description' field, a 'Loop Table Rows' checkbox, and an 'If' section with a 'Select Variable' dropdown and an 'Expression' field (highlighted with a red box and circle 1). Below the 'If' section are buttons for 'Add Sub Table' (highlighted with a red box and circle 2), 'Add Compound Variable', and 'Add Compound Table'. The 'Then' section has an 'Add Compound Table' button. There are several checkboxes: 'Diagnosis Note' (checked), 'Set as Status Code for This Device' (unchecked), 'Status Code for Network Intent' (unchecked), 'Export the table row to CSV report' (unchecked), 'Follow-up Network Intents' (unchecked, highlighted with a red box and circle 5), and 'Break Current Loop' (unchecked, highlighted with a red box and circle 3). At the bottom of the configuration area are '+ Add Elseif' (highlighted with a red box and circle 4) and '+ Add Else' buttons. The bottom of the window shows 'Help | All Variables', 'Cancel', and 'Apply' buttons.

2.2.1 Support Inline Expression

Users can use the inline expression in the definition of a diagnosis instead of defining a compound variable and then using that variable in the diagnosis.

The inline variable expression supports the simple math operation (+, -, *, and /) and built-in functions, which are expanded to manipulate the string, IP address, MAC address, interface name, etc.

The screenshot displays the 'CLI Command Diagnosis' window for device 'US-BOS-R1'. The interface is divided into three main sections: '1. Define Baseline', '2. Define Diagnosis', and '3. Debug'. The 'Define Diagnosis' section is active, showing a table of baseline commands and their corresponding diagnosis configurations.

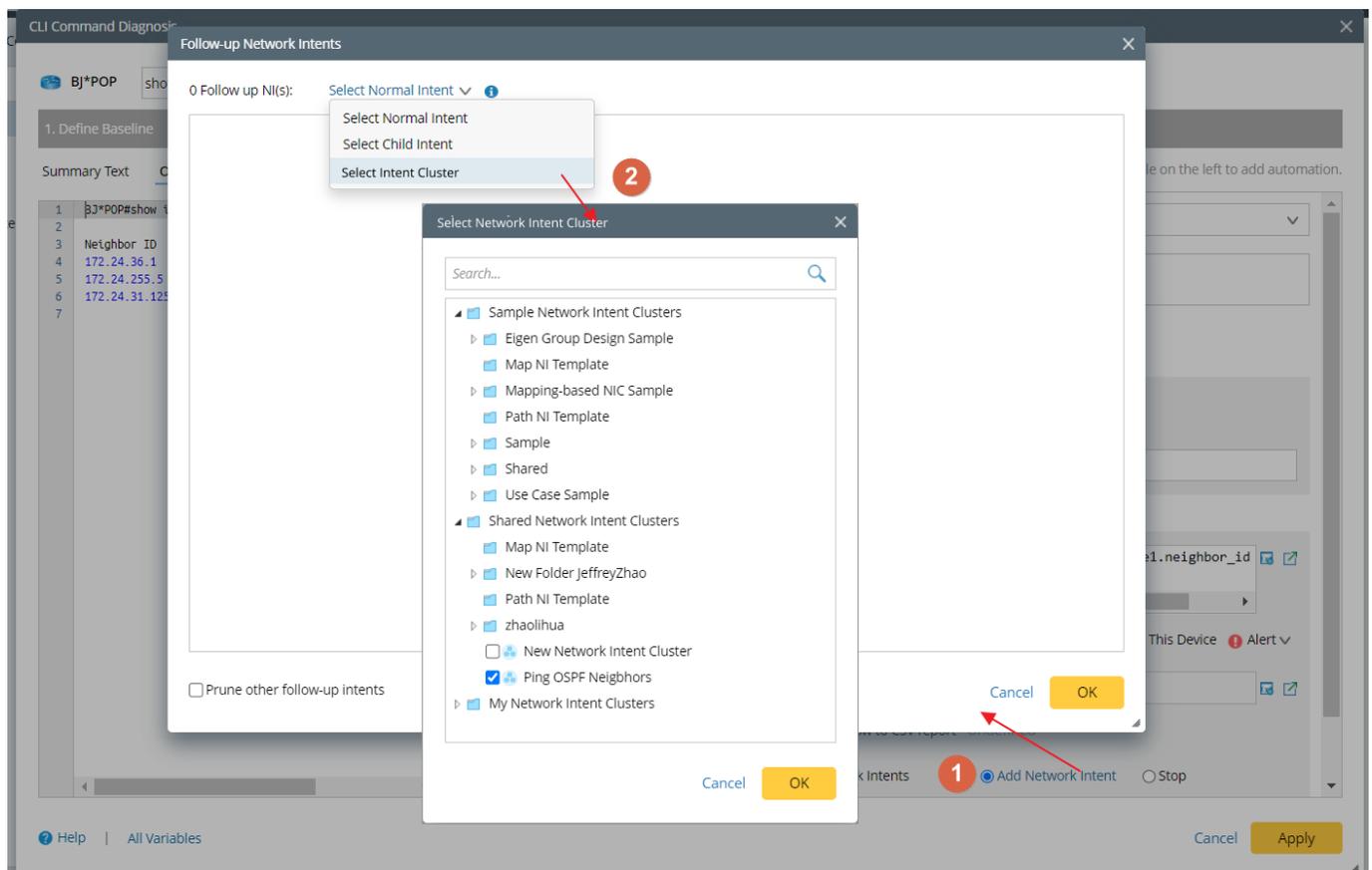
Summary Text	Original Text
1 Topology-MTID	Cost Disabled Shutdown
2 0 1 no no	OSPF Nbr Down
12 Enabled by interface config, including	
13 Loopback interface is treated as a	
22 Tunnel0 is up, line protocol is up	
23 Internet Address 10.8.1.65/30, Area	
24 Process ID 1, Router ID 10.11.11.11	
25 Shutdown	
26 0 65 no	
28 Transmit Delay is 1 sec, State POINT	
29 Timer intervals configured, Hello 1	
30 oob-resync timeout 40	
31 Hello due in 00:00:01	
32 Supports Link-local Signaling (LLS)	
33 Cisco NSF helper support enabled	
34 IETF NSF helper support enabled	
35 Index 2/2, flood queue length 0	
36 Next 0x0(0)/0x0(0)	
37 Last flood scan length is 0, maximum is 0	
38 Last flood scan time is 0 msec, maximum is 0	
39 Neighbor Count is 0, Adjacent neighbor count	
40 Suppress hello for 0 neighbor(s)	
41 Ethernet0/1 is up, line protocol is up	
42 Internet Address 10.8.1.53/29, Area 0, Attach	

An 'Expression Using Selected Variable' dialog box is open, showing the variable 'US-BOS-R1 \$cpu_usage' and the expression 'StringToNumber(Baseline(\$cpu_usage)) * 2'. The dialog has 'Cancel' and 'OK' buttons.

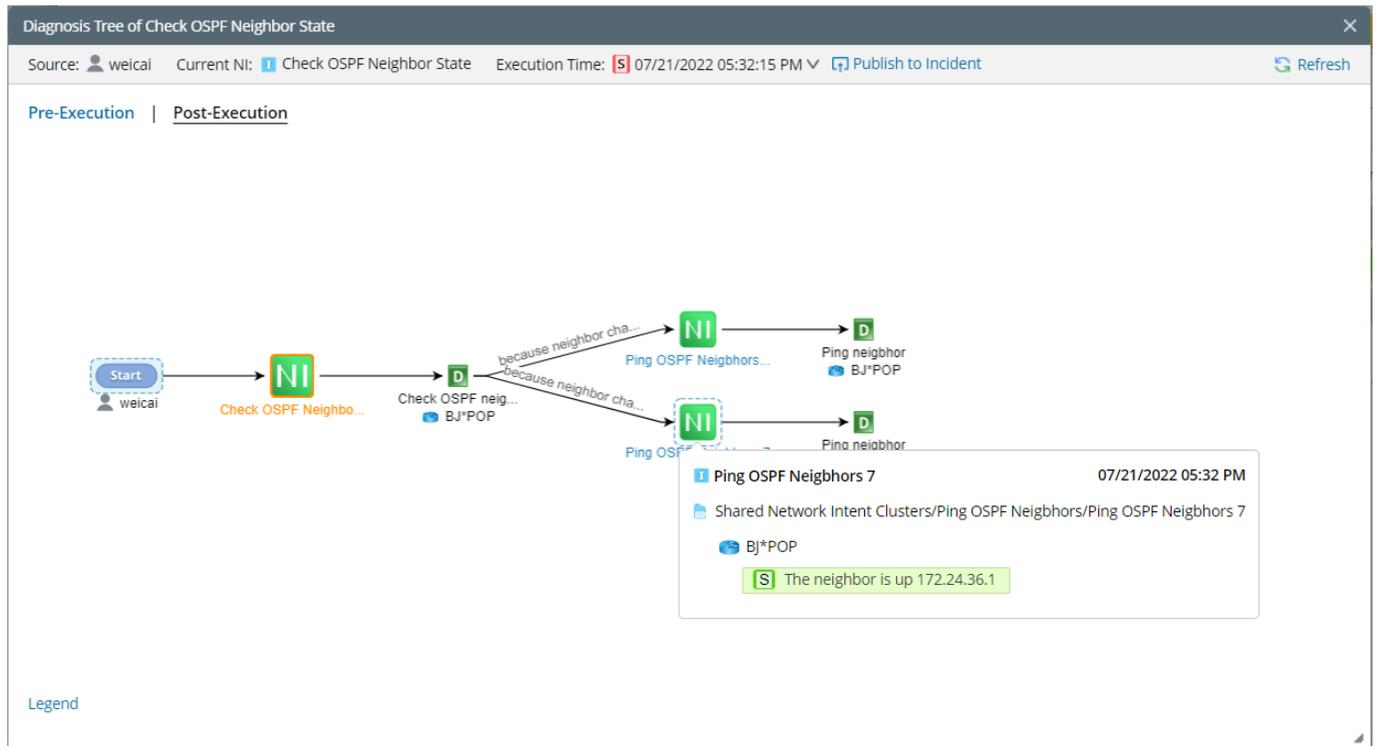
The background interface shows the 'Define Diagnosis' configuration for 'OSPF Nbr Down'. It includes fields for 'Name' (OSPF Nbr Down), 'Boolean Expression' (A), and 'Then' actions (Diagnosis Note, Set as Status Code for This Device, Status Code for Network Intent).

2.2.2 Support NIC as the follow-up NI

In the real world, much of the diagnosis logic is done at the interface/port level, and users could have many interface-based NI per device. Therefore, the follow-up NI should be dynamically selected from a NIC by parameters to fill the gap that NI cannot use the parameter to drive the “next-step” execution. For example, for each entry changed in OSPF neighbors, a user wants to do a “Ping to state changed neighbor” in follow-up NIC with macro variable and pass the obtained value to command: `ping $neighbor_ip`.



The matched member NIs in the follow-up NIC will be displayed in the Diagnosis Tree (Post-Execution).



2.2.3 Support Elseif branch in diagnosis

R11 adds the **Elseif** branch under the **If** branch of the diagnosis. For example, it can be used to judge whether the subnet of the BGP route in the route summary has changed. If the subnet has not changed, check whether the subnet of the OSPF route has changed.

CLI Command Diagnosis

BJ_Acc_SW6 show ip route summary Retrieve Live Data Last Updated: 08/05/2022 01:33:12 PM

1. Define Baseline 2. Define Diagnosis 3. Debug

Summary Text Original Text Search...

	route_source	Networks	Subnets	Replicates	Overhea
4	route_source				
5	connected	0	11	0	748
6	static	1	1	0	136
7	application	0	0	0	0
8	ospf 1	0	58	0	4284
11	nhrp	0	0	0	0
12	bgp 65001	0	5	0	340
14	internal	17			
15	Total	18	75	0	5508

route change

Add Note Add Diagnosis Can also click a variable on the left to add automation.

ElseIf Delete

A BJ_Acc_SW6 Current
route_source Contains ospf

B BJ_Acc_SW6 Current BJ_Acc_SW6 Baseline
subnets Does not ... subnets

C Select Variable

Boolean Expression: A and B

Then

Diagnosis Note: (BJ_Acc_SW6.route_summary) route changed.

Set as Status Code for This Device Alert

Status Code for Network Intent Alert

Export the table row to CSV report Undefined

Follow-up Network Intents Add Network Intent Stop

Break Current Loop

+ Add Elseif + Add Else

Help All Variables Cancel Apply

2.2.4 Better support for the table-based diagnosis

R11 provides better support for the table-based diagnosis, including:

- The **Whole Table-based Diagnosis** to compare the current table with its last or baseline value to detect changes.

When the Loop Table Rows option is not selected, a user can select a table, an operator (Equals, Does not equal, Is empty, and Is not empty), and the table to be compared.

The screenshot shows the 'Command Diagnosis' interface. At the top, the command 'show ip ospf interface' is entered for device 'US-BOS-R2'. The interface is divided into three steps: '1. Define Baseline', '2. Define Diagnosis', and '3. Debug'. The 'Define Diagnosis' step is active.

The left pane shows the command output for 'show ip ospf interface'. The output includes BGP table version information, status codes, origin codes, and a table of routes. A 'Check route' button is visible next to the first route entry.

Network	Next Hop	Metric	Lo
*> 1.1.1.246/32	10.8.1.49	32	
*> 10.2.2.2/32	10.8.1.49	21	
*> 10.3.3.3/32	10.8.1.49	22	
*> 10.8.0.0/16	0.0.0.0	11	
*> 10.8.1.0/28	10.8.1.49	21	
*> 10.8.1.32/29	10.8.1.49	20	
*> 10.8.1.48/29	0.0.0.0	0	
*> 10.8.1.64/30	10.8.1.53	75	
*> 10.8.1.240/32	10.8.1.53	11	
* i 10.8.2.0/30	10.11.11.11	0	
*>	10.99.1.52	20	
* i 10.8.2.4/30	10.11.11.11	0	
30000 65001 ?			
*>	10.99.1.52	20	
* i 10.8.2.8/30	10.11.11.11	0	
*>	10.99.1.52		
* i 10.8.2.12/30	10.11.11.11		
*>	10.99.1.52	20	
* i 10.8.2.8/30	10.11.11.11	0	

The right pane shows the configuration for the diagnosis rule. The 'If' section is set to 'Loop Table Rows'. The condition is defined as 'A' (US-BOS-R1) with the variable 'bgp_routes_to_critical' compared to 'B' (Select Variable) using the operator 'Does not equal'. The 'Then' section includes options for adding a diagnosis note, outputting a summary message, setting as a status code for the device, and following up on network intents.

- Set the table key in the diagnosis

The screenshot shows the 'Command Diagnosis' interface. On the left, the 'Original Text' pane displays the output of the command 'show ip bgp' on device US-BOS-R1. The output includes a table with columns 'Network', 'Next Hop', and 'Metric'. The table contains 28 rows of route information. A 'Check route' button is visible next to the first row of the table.

On the right, the 'Define Diagnosis' pane is active. It shows a configuration for a table-based diagnosis. The 'If' condition is set to 'Loop Table Rows' with the table key set to 'network'. A dropdown menu is open, showing options: 'f', 'network' (selected), 'next_hop', 'metric', 'locprf', 'weight', and 'path'. The 'Then' section is also visible, with a 'Diagnosis Note' set to 'The network "10.8.1.32/29" exists in the BGP route table.' and other options like 'Set as Status Code for This Device' and 'Break Current Loop'.

- Support the sub table creation

R11 supports defining a sub table to define a table-based batch comparison diagnosis, which can reduce the complexity of creating a table-based diagnosis. The sub table can be created by either keeping or removing the value at the specific columns, which can be used to compare only the desired data in a table. For example, users only want to analyze the route entries related to the critical application subnet in the device route table and ignore the changes of other dynamic routes, reducing the noise alert that may appear in diagnosis.

The screenshot displays two overlapping windows from a network management tool. The primary window is titled 'Add Sub Table (Filtering Row)'. It features a 'Table Name' field set to 'bgp_routes_critical', a 'Base Table' dropdown set to 'bgp_routes', and a 'Filtering Logic' section. The logic is set to 'Only Keep' table rows where the value of 'network' exists in a predefined global list named 'critical_subnet'. Below this, a table lists the 'critical_subnet' entries:

subnet(network)	hosting_device	description
1.1.1.246/32	US-BOS-R2	This route is to PE1
10.2.2.2/32	US-BOS-R3	This route is to PE2
10.3.3.3/32	US-BOS-R4	This route is to PE3

A 'Calculate' button is present. Below, two tables are shown: 'Base Table (bgp_routes)' and 'New Table (bgp_routes_critical)'. The new table contains the following data:

f	network	next_hop	metric	locprf	weight	path	description
*>	1.1.1.246/32	10.8.1.49	32		32768	?	This route is to PE1
*>	10.2.2.2/32	10.8.1.49	21		32768	?	This route is to PE2
*>	10.3.3.3/32	10.8.1.49	22		32768	?	This route is to PE3
*>	10.8.0.0/16	0.0.0.0	11		32768	?	ThisAN edge
*>	10.8.1.0/28	10.8.1.49	21		32768	?	This route is to core switch

The secondary window, titled '3. Debug', shows a tree view of network objects. A red arrow points to the 'Add Sub Table' option in the tree. A message box in the debug window states 'route table no changed.'.

- Support the global network data tables

R11 supports predefining **Global Network Data Tables** as the data source for Sub Table to define white or block list. It can also be used for **Mapping Macro Variable** in NIT definition.

The screenshot shows the 'Domain Management' interface in NetBrain. The top navigation bar includes 'Tenant: Initial Tenant', 'Domain: i690_domain', 'Admin', 'Operations', and the NetBrain logo. The main content area is titled 'Global Network Data Tables' and features a search bar, '+ Add Column', '+ Add Row', and 'Import from CSV' options. A table with three columns is displayed: 'subnet', 'hosting_device', and 'description'. The table contains seven rows of data. A 'Submit' button is located at the bottom right of the table area.

subnet	hosting_device	description
1.1.1.246/32	US-BOS-R2	This route is to PE1
10.2.2.2/32	US-BOS-R3	This route is to PE2
10.3.3.3/32	US-BOS-R4	This route is to PE3
10.8.0.0/16	US-BOS-R5	This route is to WAN edge
10.8.1.0/28	US-BOS-R6	This route is to core switch
11.8.1.32/29	US-BOS-R7	This route is to ACI edge

- Break table loop

A new Output (Break Table Loop) can be added to the diagnosis output. Break Table Loop can better control the flow of the loop table row. When the table row loops, the current loop table will be

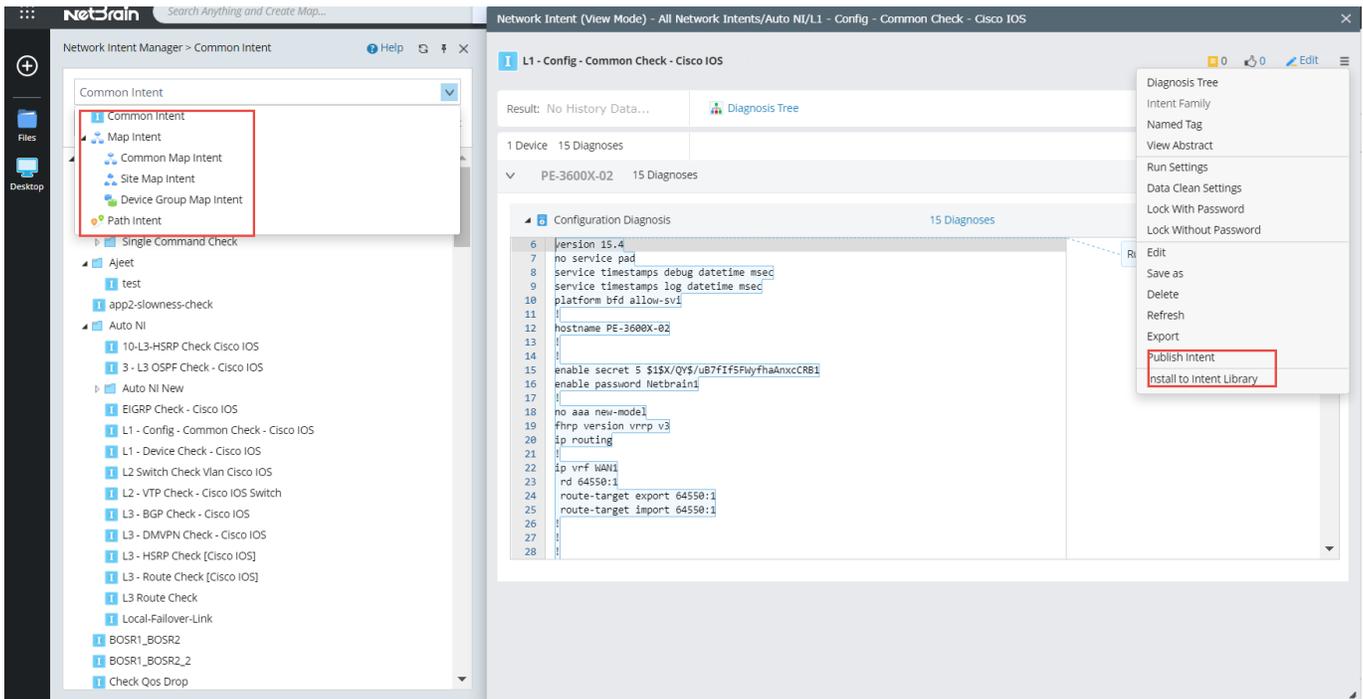
terminated to reduce the calculation of the redundant loop if the condition is satisfied.

The screenshot displays the 'Command Diagnosis' window for the command 'show ip ospf interface' on device US-BOS-R2. The interface is divided into three main sections: '1. Define Baseline', '2. Define Diagnosis', and '3. Debug'. The 'Define Baseline' section shows the original command output, which includes a BGP table with columns for Network, Next Hop, and Metric. The 'Define Diagnosis' section is active, showing a configuration where a loop table row is selected. The configuration includes an 'If' condition: 'US-BOS-R1 Current' with 'network' equals '10.8.1.32/29'. The 'Then' section is configured to trigger a 'Diagnosis Note' (green) with the message 'The network \"10.8.1.32/29\" exits in the BGP route table.' and to 'Set as Status Code for This Device' (Success). The 'Break Current Loop' option is checked and highlighted with a red box. A 'Check route' button is visible next to the selected row in the BGP table.

Network	Next Hop	Metric
*> 1.1.1.246/32	10.8.1.49	32
*> 10.2.2.2/32	10.8.1.49	21
*> 10.3.3.3/32	10.8.1.49	22
*> 10.8.0.0/16	0.0.0.0	11
*> 10.8.1.0/28	10.8.1.49	21
*> 10.8.1.32/29	10.8.1.49	20
*> 10.8.1.48/29	0.0.0.0	0
*> 10.8.1.64/30	10.8.1.53	75
*> 10.8.1.240/32	10.8.1.53	11
* i 10.8.2.0/30	10.11.11.11	0
*>	10.99.1.52	20
* i 10.8.2.4/30	10.11.11.11	0
30000 65001 ?		
*>	10.99.1.52	20
* i 10.8.2.8/30	10.11.11.11	0
*>	10.99.1.52	20
* i 10.8.2.12/30	10.11.11.11	0
*>	10.99.1.52	20
* i 10.8.2.8/30	10.11.11.11	0

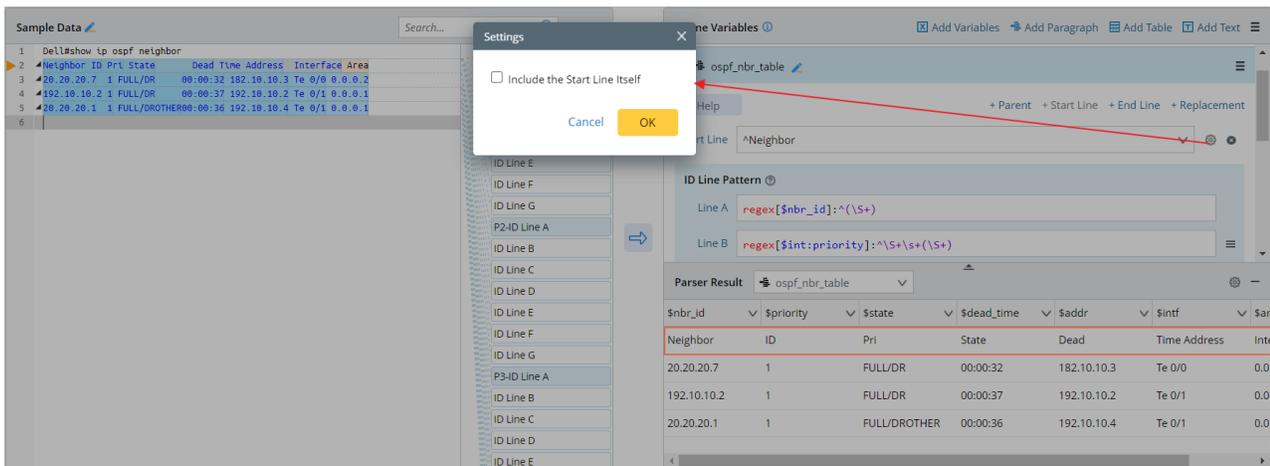
2.3 Intent Manager

R11 provides a standalone Intent Manager to manage all intents, including common intents, Map Intents and Path Intents. Besides editing, creating, and deleting functions, users can publish and install an intent to the library, two new intent functions.



2.4 Visual Parser Enhancements

NetBrain R11 supports the setting **Include the Star Line Itself** for the Start Line and **Include the End Line Itself** for the End Line, determining whether the Start Line and the End Line are included in the matched text.



During the Collector Parser definition, the `<% $var %>` is supported in the LinesByKeyword expression syntax in Variable Line. The specific var1 in syntax `<% $var1 %>` refers to the ID variable defined in the ID Line. With this improvement, users can define multiple lines of configlets simultaneously.

```

Sample Data
28
29
30
31 64412 message lines logged
32 ort 514, audit disabled,
33
34 d,
35 teg,
36 y=ID,
37 mbr disabled
38
39 ort 514, audit disabled,
40
41 d,
42 teg,
43 y=ID,
44 mbr disabled
45
46 VRF Name:
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Define Variables

Collector1

Help + Parent + Start Line + End Line + Replacement

ID Line mismatch discovered on \$interface

Variable Line Pattern

Line 1 LinesByKeyword[\$all_log]: mismatch discovered on <% \$interface %>

Parser Result Collector1

Interface	\$all_log
Ethernet0/3	*Sep 9 12:36:29.849: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on E...
Ethernet0/1	*Sep 9 12:37:14.856: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on E...

3 Replicate Intent

Replicating the logic of a NI (seed NI) to the whole network is essential to build PDAS on a large scale. R11 introduces a simpler method, **Intent Cluster (NIT)**, to replicate the intent while improving the Intent Cluster (NIC), adding the **Auto Mode** to reduce 7 steps to 3 steps and the **Auto Test** feature to simplify the definition of the device classifier and eigen grouping and create the member NIs only for the qualified devices.

3.1 NIC Improvements

3.1.1 Auto Mode

The Auto Mode (enabled by default) reduces NIC definition from 7 steps to three simple steps:

The screenshot displays the 'New Network Intent Cluster' configuration window. The process flow is summarized as follows:

- 1. Input Devices (all devices)**: The starting point, leading to both 'Seed NI' and 'Class'.
- 2. Seed NI (1 seed device)**: A step that branches from 'Input Devices' to 'Seed NI' and 'Class'.
- 3. Seed Logics (1 seed logic)**: A step that branches from 'Seed NI' to 'Logic' and 'Eigen'.
- 6. Target Seed (1 criterion)**: A step that branches from 'Logic' and 'Eigen' to 'Match'.
- 7. Member NI (1 member NI)**: The final output, resulting from the 'Match' step.

The 'Input Devices Settings' panel is shown with 'Auto Mode' checked. The 'Select Grouping Method' dropdown is open, showing options: 'By Site', 'By Device Group', 'By Device', 'By Path', and 'By Map'. The 'Per Device' dropdown is also open, showing options: 'Per Device', 'Per VLAN Group', 'Per Subnet', 'Device and Its L3 Neighbors', 'Device and Its L2 Neighbors', and 'All in One Group'. A green arrow points to the 'Per Device' option.

The 'View Device Data' panel shows a list of 30 items of 30 devices. The list includes:

Hostname
AS30000
R20
US-BOS-R1
US-BOS-SW2
US-NYC-Enterprise-SW1
US-LAX-FortiGate-FW
US-SMF-R2
US-BOS-R2
R3

1. Select the input devices.

R11 adds two ways to select the input devices: **by path** and **by map**. When users select inputting the device **by device**, they must select the method to create the group, which can be **per device, per VLAN group, per subnet, device and L3 neighbors, device and its L2 neighbors, and all in one group**.

2. Select Seed NIs.

The auto mode only supports the single device diagnosis. The system will ask users to disable the auto mode if the Seed Intent contains a cross-device diagnosis.

3. Create the member NIs.

The member NIs will be created by the type of the input devices or the method to create the group:

- **By map:** all devices on the same map will belong to a member NI.
- **By Site:** all devices of a site will belong to a member NI.
- **By Device Group:** all devices in a device group will belong to a member NI.
- **By Path:** all devices in a path will belong to a member NI.
- **By Device:**
 - **Per device:** a member NI will be created for each device.
 - **Per VLAN group:** a member NI will be created for all devices belonging to a VLAN group.
 - **Per subnet:** a member NI will be created for all devices belonging to a subnet.
 - **Device and L3 neighbors:** a member NI will be created for the device and its L3 neighbors.
 - **Device and its L2 neighbors:** a member NI will be created for the device and its L2 neighbors.
 - **All in one group:** only one member NI is created to include all devices.

The system automatically creates other nodes. Users can disable the auto nodes and edit these nodes.

NIC auto mode can be used to batch create the map intents (input the devices by map so that the map name will be used as eigenvalue) and Path intents (input the devices by path so that the path name will be used as eigenvalue).

3.1.2 Auto Test

R11 adds an option, **Test Seed NI variable**, to node **Target Seed**. With this option enabled, users can select the **seed NI variables**. If one of these variables is not retrieved or parsed successfully from a device, the system will not create member NI for this device. With this option, users can create the member NIs that do not create meaningful results and simplify the definition of Device Classifiers (node 4) and Group by Eigen Values (node 5) when multiple vendors or commands are involved.

Seed NI Variable Setting

Please select Seed NI Variables for testing:

- route change monitoring
 - D1 [R1]
 - show ip route
 - parent
 - route
 - route_table (selected)
 - age
 - dist1
 - dist2
 - next_hop
 - route

Cancel OK

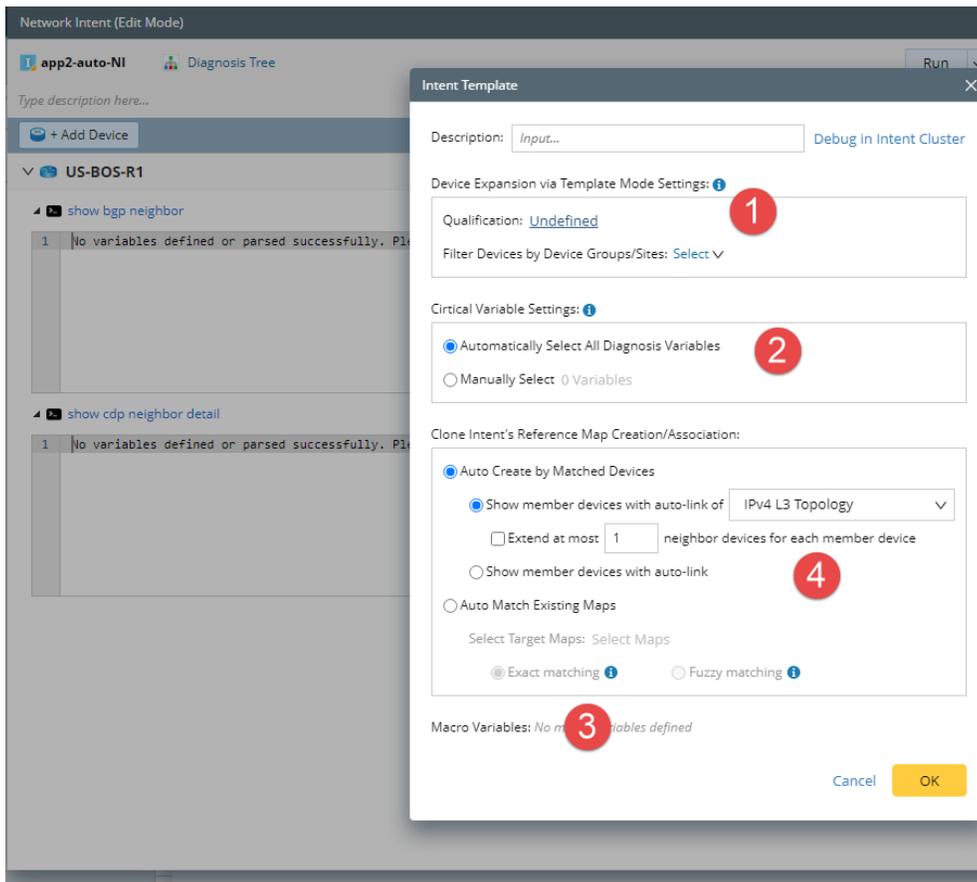
Eigen Group	Eigen Value
Group 1	(US-SMF-SW2)
Group 2	(US-NYC-R1)
Group 3	(US-NYC-Enterprise)
Group 4	(US-NYC-DMZ-SW1)
Group 5	(US-BOS-FW/act)
Group 6	(US-BOS-R1)
Group 7	(US-SMF-R2)

3.2 Intent Template (NIT)

R11 further simplifies the replication of the intent by inventing the **Intent Template (NIT)**. A user can define NIT for any NI with the single-device diagnosis. Like NIC auto mode, if an Intent contains cross-device diagnosis, it cannot be replicated by NIT.

3.2.1 Define NIT

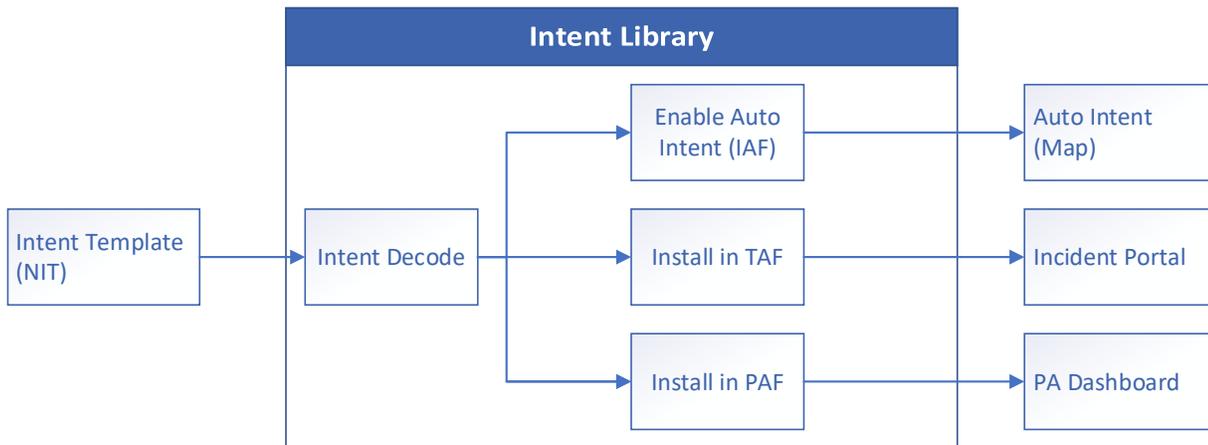
Users can define an NIT in four easy steps:



1. Define the qualified devices: Cisco IOS devices with OSPF configuration. Further, you can filter devices by device group or sites.
2. Define critical variable settings: The system selects all diagnosis variables by default. You can manually set a subset of the variables. A member NI will only be created for a device if all critical variables are retrieved and parsed successfully.
3. If NI has macro variables, you may define macro variables.
4. Define how to create the reference map.

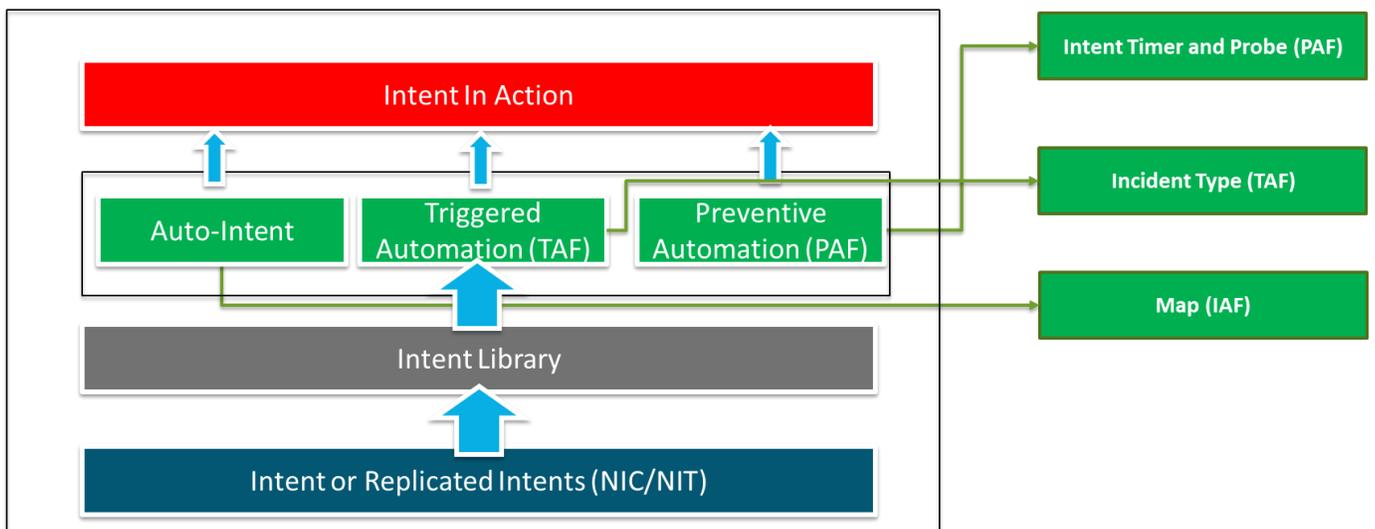
3.2.2 Install and Use NIT

After NIT is defined, it can be installed and used throughout the system. First, you must instruct the system to decode the intent, a process to decide whether to create the member NI based on the template definition. Then, you can enable the intent to be displayed in the map under the [auto intent](#); install it in [TAF](#) so that it can be triggered; install it in [PAF](#) so that the Probe can trigger it. R11 add [Intent Library](#) as a central console for these operations.



4 Install Intent

Intent or replicated Intent (NIT or NIC) can be executed manually from the map or path (Interactive Automation, IAF), triggered by a 3rd party system (Triggered Automation, TAF), or triggered by the Probe (Preventive Automation, PAF).

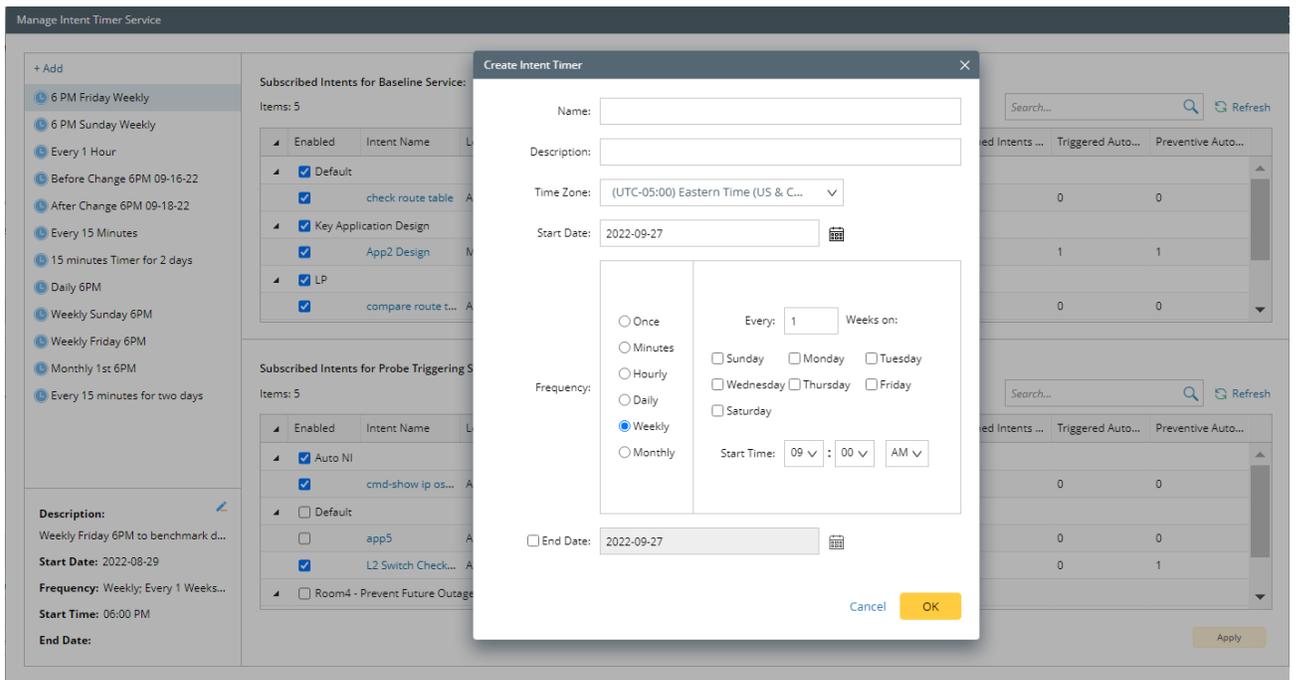


4.1 Intent Library

The intent library is the central console to install an intent:

The screenshot displays the 'Intent Library' interface. At the top, there are tabs for 'Installed Intents', 'NetBrain Download', and 'Published Intents'. Below the tabs, there is a search bar and a filter dropdown set to 'All'. A table lists various intents with columns for Intent Name, Location, Intent Mode, Intent Baseline, Intent Decoding, Auto Intent, Cloned Intents, Triggered Automations, and Preventive Automations. The 'specific route check' intent is highlighted. Below the table, a configuration panel for the 'specific route check' intent is shown. This panel includes sections for 'Intent Decoding' (with 'Recurring Decode' and 'One-Time Decode' options), 'Intent Timers' (with 'Update Baseline by Timer' and 'Execute Intent by Timer' options), and 'Triggered Automation' (with a table of incident types). Red circles 1 through 5 are overlaid on the interface to highlight specific features: 1. The '+ Add Intent' button, 2. The 'Recurring Decode' radio button, 3. The 'Update Baseline by Timer' dropdown menu, 4. The 'Enable Auto Intent' checkbox, and 5. The 'Application Check' incident type in the automation table.

1. Add an intent as the standalone or the template.
2. Configure the intent decoding if the intent is installed as a template. Users can select one-time or recurring decoding. The NIT decoding service creates a list of qualified devices based on the NIT setting. These qualified devices will be used later to clone NIs on the fly.
3. Configure Intent Timers. Configure the timer to update the intent baseline and execute the intent. The execution timer is used by PAF, which has two options, **Execute by Timer** (always execute when the timer is up) and **via Probe** (only when the condition of Probe is satisfied).



4. Enable [auto intent](#) so the intent decode service can clone intents for the map.
5. [Install NI/NIT to PAF](#) and [TAF](#).

Besides the installed intent, NetBrain downloaded tasks and published intents are also displayed in the Intent Library.

NetBrain team continuously creates the intents for common network problems and publishes them through Knowledge Cloud, which can be automatically downloaded to your system. The downloaded files include a configuration Orchestration File for each task, which can be executed to download and install the related intents.

Installed Intents **NetBrain Download** Published Intents

Library: VL Library1 Published at: 9/9/2022, 12:20:00 PM Install Cisco APT, NIT, DVT

View: All Status Search...

Sequence	Task	Prerequisite	Last Status	Last Execution Time ...	Executor	Task2 - Install NI																
1	Email All_ChangeFolderName1					Description: This is for the installation NI.																
1.1	Neighbor Check Dynamic		None																			
1.2	Neighbor Check Static		None																			
2	DepEmail inSameFdr_ChangeFolderName2																					
2.1	Task1 - CopyMulti-NI		None																			
2.2	Task2 - Install NI	0 of 2 Installed	None																			
2.3	Task3 - Install VIT_EmailTrigger_TestDynamic11	0 of 2 Installed	None																			
2.4	Task4 - Install IT	0 of 3 Installed	None																			
2.5	Task5 - Install TAF	0 of 4 Installed	None																			
2.6	Task6 - TAF_Operator		None																			
2.7	Task7 - TAF_Operator1		None																			
3	DepEmail inDiffFdr1					<p>Logs Associated Assets</p> <p>Filter: All</p> <table border="1"> <thead> <tr> <th>Time</th> <th>Messages</th> </tr> </thead> <tbody> <tr> <td>9/29/2022, 4:07:05 AM</td> <td>Start installing task Task2 - Install NI, process id is 17528.</td> </tr> <tr> <td>9/29/2022, 4:07:05 AM</td> <td>Start the validation operation of Asset Config File Asset/NIT by jerry21.ninstall.</td> </tr> <tr> <td>9/29/2022, 4:07:05 AM</td> <td>Successfully validated all Asset Config files of INSTALL_INTENT.</td> </tr> <tr> <td>9/29/2022, 4:07:05 AM</td> <td>Start the installation operation of Asset Config File Asset/NIT by jerry21.ninstall.</td> </tr> <tr> <td>9/29/2022, 4:07:05 AM</td> <td>Start loading Asset Config Assets/NIT by jerry21.ninstall</td> </tr> <tr> <td>9/29/2022, 4:07:05 AM</td> <td>Start installing Intent All Network Intents/NB Download/Qerry/NIT by jerry21</td> </tr> <tr> <td>9/29/2022, 4:07:05 AM</td> <td>Successfully installed or updated Intent All Network Intents/NB Download/Qerry/NIT t</td> </tr> </tbody> </table>	Time	Messages	9/29/2022, 4:07:05 AM	Start installing task Task2 - Install NI, process id is 17528.	9/29/2022, 4:07:05 AM	Start the validation operation of Asset Config File Asset/NIT by jerry21.ninstall.	9/29/2022, 4:07:05 AM	Successfully validated all Asset Config files of INSTALL_INTENT.	9/29/2022, 4:07:05 AM	Start the installation operation of Asset Config File Asset/NIT by jerry21.ninstall.	9/29/2022, 4:07:05 AM	Start loading Asset Config Assets/NIT by jerry21.ninstall	9/29/2022, 4:07:05 AM	Start installing Intent All Network Intents/NB Download/Qerry/NIT by jerry21	9/29/2022, 4:07:05 AM	Successfully installed or updated Intent All Network Intents/NB Download/Qerry/NIT t
Time	Messages																					
9/29/2022, 4:07:05 AM	Start installing task Task2 - Install NI, process id is 17528.																					
9/29/2022, 4:07:05 AM	Start the validation operation of Asset Config File Asset/NIT by jerry21.ninstall.																					
9/29/2022, 4:07:05 AM	Successfully validated all Asset Config files of INSTALL_INTENT.																					
9/29/2022, 4:07:05 AM	Start the installation operation of Asset Config File Asset/NIT by jerry21.ninstall.																					
9/29/2022, 4:07:05 AM	Start loading Asset Config Assets/NIT by jerry21.ninstall																					
9/29/2022, 4:07:05 AM	Start installing Intent All Network Intents/NB Download/Qerry/NIT by jerry21																					
9/29/2022, 4:07:05 AM	Successfully installed or updated Intent All Network Intents/NB Download/Qerry/NIT t																					
3.1	Task1 - CopyMulti-NI		Installed	9/29/2022, 4:07:05 AM	zhouljg1																	
3.2	Task2 - Install NI	1 of 1 Installed	Installed	9/29/2022, 4:07:05 AM	zhouljg1																	
3.3	Task3 - Install VIT_EmailTrigger_TestDynamic21	2 of 2 Installed	Installed	9/29/2022, 4:07:07 AM	zhouljg1																	
3.4	Task4 - Install IT	3 of 3 Installed	Installed	9/29/2022, 4:07:07 AM	zhouljg1																	
5	DepEmail inDiffFdr2																					
5.1	Task5 - Install TAF	4 of 4 Installed	Installed	9/29/2022, 4:07:07 AM	zhouljg1																	
6	All FeatureData																					
6.1	Task1 - Copy NI		Installed	9/29/2022, 4:07:07 AM	zhouljg1																	
7	pid server = 2																					
7.1	Task1 - Copy NI		None																			

4.2 Auto Intent

When users open a map, the automatically decoded Intents for the map devices are listed under the Auto Intents by device or intent. Users can select one or multiple auto intents:

The screenshot shows the Intent interface with the following components:

- Left Panel (Auto Intent):** A list of auto intents categorized by device. A red circle '1' highlights the '1-Weekly-Config-Analysis (5)' intent.
- Center Panel (Run Intent):** A configuration editor for the selected intent. A red circle '2' highlights the 'Create' button, and a red circle '3' highlights the 'Run Intent' button. The configuration shows a device diagnosis for 'US-BOS-R2' with a configuration snippet including 'service password-encryption', 'hostname US-BOS-R2', and 'clock timezone EET 2 0'.
- Right Panel (Map):** A network map showing devices like 'US-NYC-R1', 'MPLS', 'CA-TOR-SW2', and 'CA-TOR-R1' connected by links. A red circle '3' also highlights a 'Run' button on the map.

1. Select and view the auto intents.
2. Create or recreate the selected intents. The intent decoding services (configured in the **Intent Library**) do not create the intents. Instead, the auto intents are created here.

3. Run the selected auto intents. The results can be viewed in the **post-execution Diagnosis Tree**.

Diagnosis Tree of Created Via Auto Intent

Source: guangdong.liao@netbraintech.com Current NI: Created Via Auto Intent Execution Time: 09/28/2022 08:45:26 AM

Pre-Execution | Post-Execution

Start → Created Via Auto Intent (NI) → US-BOS-SW1

- Check Interface C... Configuration
- Check Interface C... Configuration
- WAN Link Context... Configuration
- Check ACL Config Configuration
- Check Pre-fix List ... Configuration
- Check BGP Config Configuration
- Check OSPF config Configuration

Legend

Diagnosis Details - Check Interface Config Change

US-BOS-R2 Configuration Execution Time: 09/28/2022 08:45:26 AM

Summary Text	Original Text
151	ip flow egress
152	ip ospf authentication message-digest
153	ip ospf message-digest-key 1 md5 7 05080F1C2431F5B4A
154	ip ospf mtu-ignore
155	service-policy input aos
156	}
157	interface Ethernet0/2

Config of interface Ethernet0/2 does ...

The results can also be viewed on the map by a special data view.

Map2 (Mixed) Page 1- Summary

Auto Intent Published Intents Map Intents

View By Device

US-BOS-R2 (2)

US-BOS-R2 Configuration 6 Diagnoses 2 Alerts

Interface Loopback0

ip ospf 1 area 0

Interface Loopback1

ip address 195.16.178.127 255.255.255.255

Interface Loopback2

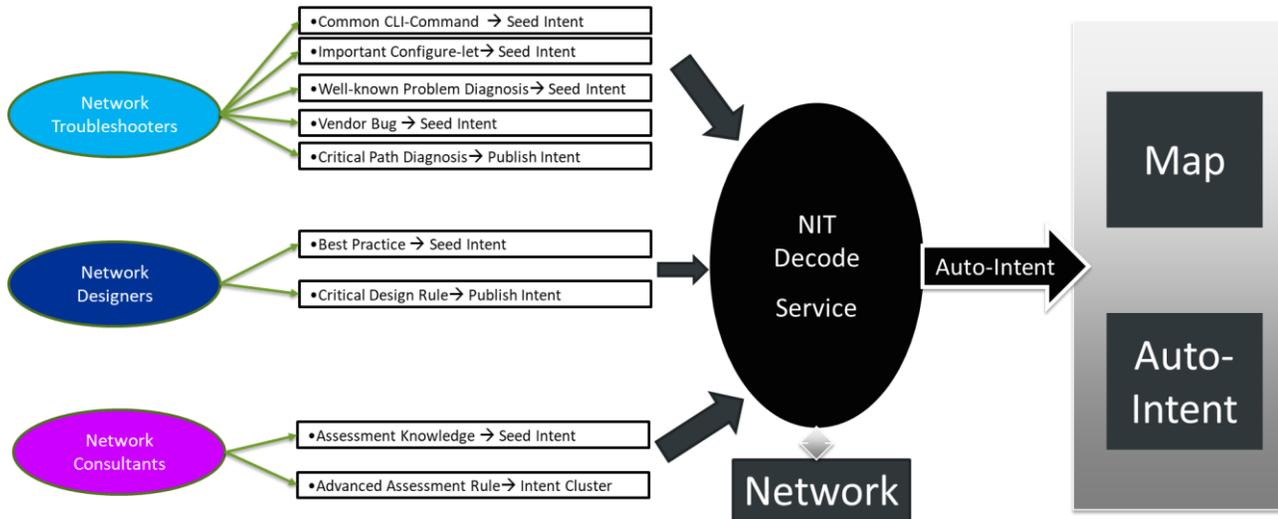
ip address 10.1.1.348 255.255.255.255

Config of interface Ethernet0/2 ...

Legend

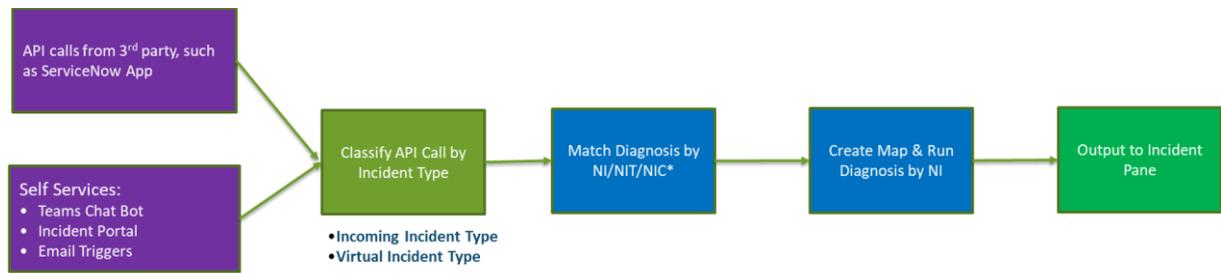
- Highlight Device
- Alert Status Code
- Success Status Code

NIT (easy to clone NI), Auto Intent decoding service, and Auto Intent can scale intents to a large network without the users' intervention.



4.3 Triggered Automation (TAF)

The triggered Automation (TAF) is a framework to connect the Automation (NI/NIT/NIC) with a network problem by the following steps:



- Receive the API calls from 3rd party system or the self-service application (Team chatbot, Emails, and Incident Portals)
- Classify the call to an Incident Type.

- Match the standalone NI or member NIs with the static data field of the API call (static diagnosis) or the hash tags (dynamic diagnosis).
- Execute NIs to create the map and/or run the diagnosis.
- Output the maps and diagnosis results to the Incident Portals.

R11 introduces a new Incident Type, Virtual Incident Type (VIT) and renames the previous Incident Type as the Incoming Incident Type. VIT encapsulates the detailed logic of Hash Tag and removes the hurdle of the diagnosis by the Hash Tag (the Dynamic Diagnosis in the previous version).

VIT has a set of data fields to describe a network incident which can be either dynamically filled via Hash Tag or statically defined.

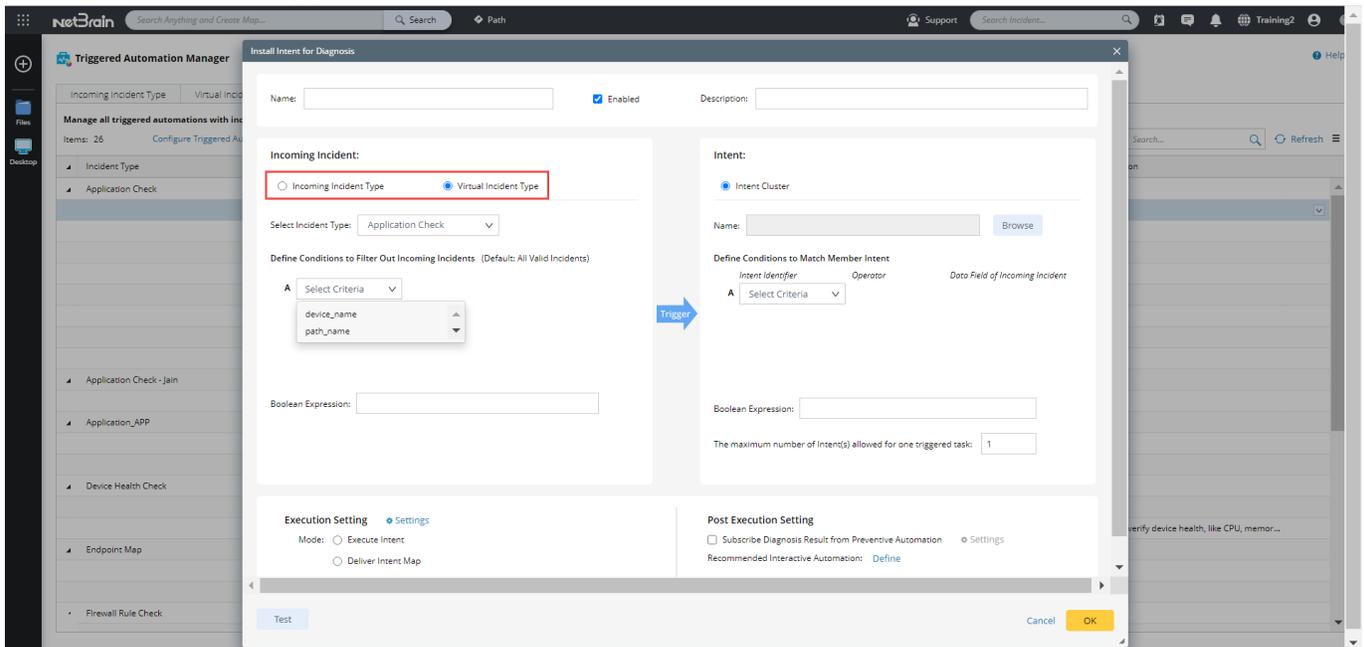
Advanced

Cancel OK

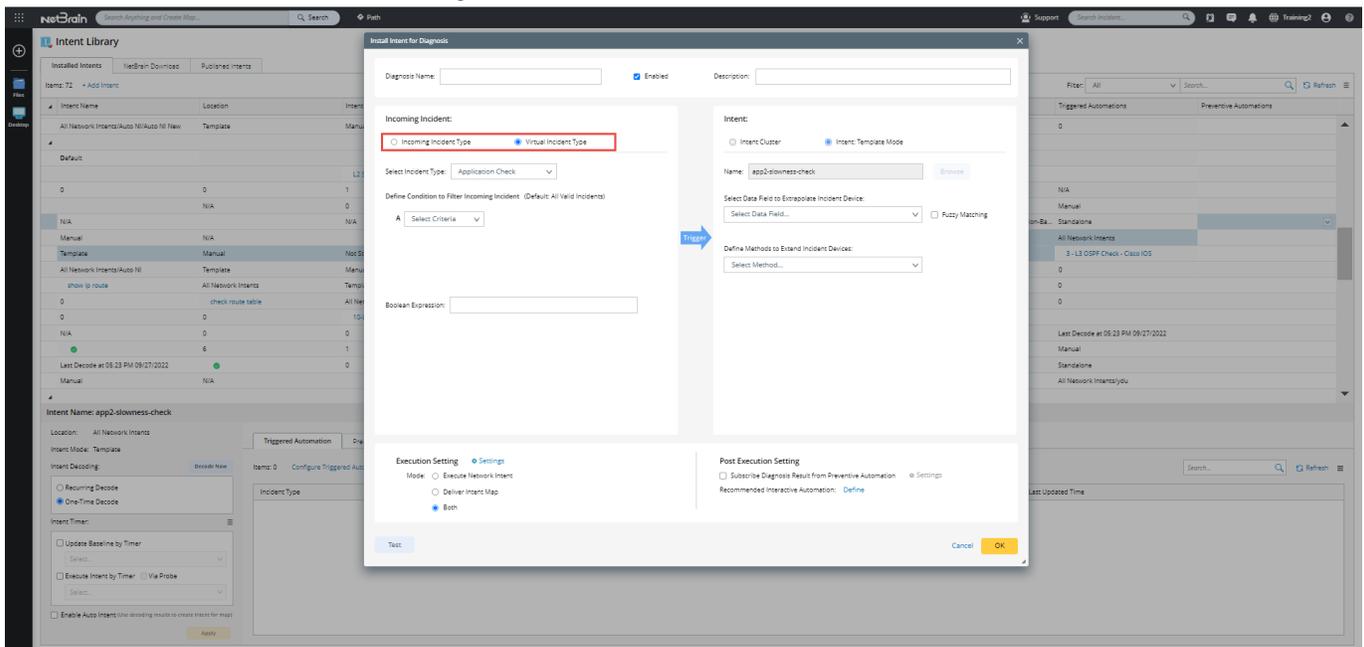
Mandatory	Field Name	Source Hash Tag Fields	Description
<input type="checkbox"/>	device_name	#_hostname	device name that path across
<input type="checkbox"/>	path_name	#_path_name	path name

The NI/NIT/NIC can be installed in both incident types with the same interface (in the current version, you can only install the **intent cluster** from the **Triggered Automation Manager** and install NI/NIC in the **Intent Library**):

- Install NIC from the **Trigger Automation Manager**:



- Install NI/NIT from the **Intent Library**:



An incoming Incident Type can be associated with one or multiple VITs and inherits the automations installed for these VITs. This one-to-many relationship can help scale TAF.

Edit Incoming Incident Type

Define complex conditions to identify data that is coming from integrated IT systems.

A Short_description Contains Jain

B Select Criteria

Boolean Expression: A

Enable this Incoming Incident Type

Associate Virtual Incident Type: ⓘ

Select Virtual Incident Types: [Select](#)

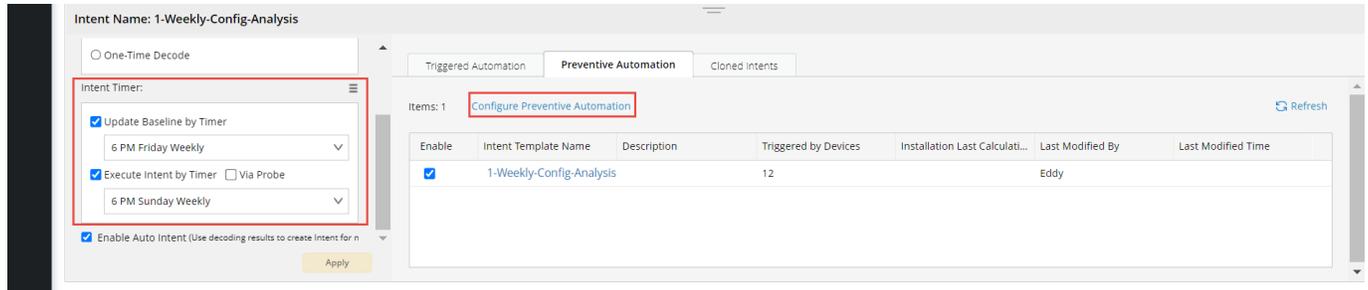
Virtual Incident Types	Data Field Type	Value Mapping	Description
Application Check - Jain	Dynamic Data Field	with Hash Tag	
BGP Device	Dynamic Data Field	with Hash Tag	Using to verify BGP status

4.4 Preventive Automation (PAF)

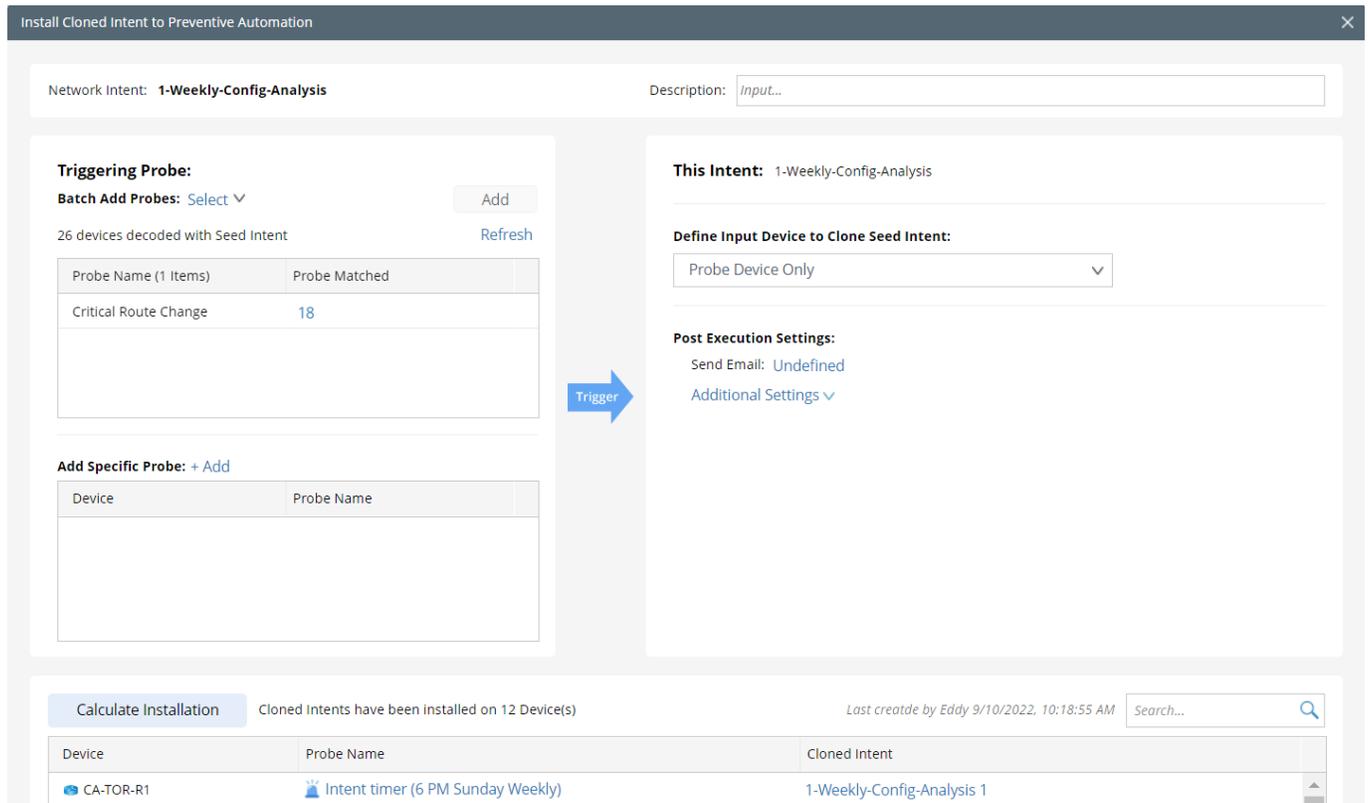
PAF uses the two-level probes (primary and secondary probes), which can further trigger any NI/NIT/NIC execution. The results are displayed in PA Dashboard.



Users can configure the timer to execute the intent for PAF, which has two options, **Execute by Timer** (always execute when the timer is up) and **via Probe** (only when the condition of Probe is satisfied).



Users can add probes to trigger this intent in the Intent Library and define the input devices to clone the seed intent.



5 Use and Execute Intent

5.1 Clone and Execute NI from Map (IAF)

The Intent Pane of a map is fully redesigned with three tabs:

- [Auto Intent](#): the automatically decoded Intents for the map devices.
- Published Intents: only listing the intents published for the map devices.
- Map Intent: an intent dedicated to this map.

The screenshot displays the NetBrain interface with the 'Intent' pane open. The 'Auto Intent' tab is selected, showing a list of commands for various devices. The 'US-LAX-R1' device is selected, showing a 'show ip route summary' command with a table of route statistics.

Route Source	Networks	Subnets	Replicates	Overh...	
connected	0	7	0	476	1260
static	0	1	0	68	180
application	0	0	0	0	0
nhnp	0	0	0	0	0
ospf 1	1	41	0	2856	7728
bgp 65001	0	1	0	68	180

5.1.1 Published Intents

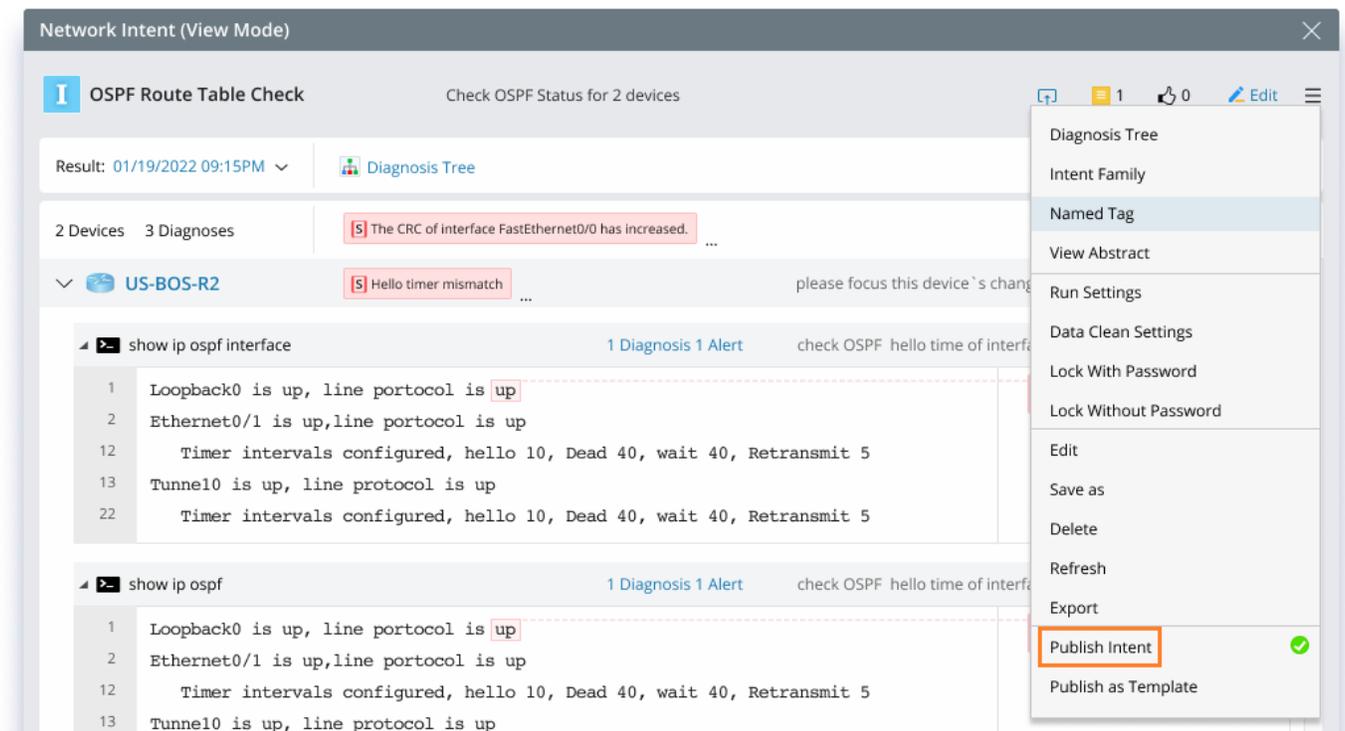
Instead of showing all intents filtered by map devices, which can be too many, R11 only shows the published intents in the map intent pane. The user can filter the published intents further by device, type, tags, and authors.

The screenshot displays a network management interface with a sidebar on the left and a main content area on the right. The sidebar contains various navigation options, including 'Filter by' (highlighted with a red box), 'Devices (11)', 'Type (3)', 'Tags (2)', and 'Authors (8)'. The main content area shows a list of 'Published Intents' (highlighted with a red box) under the 'Intent' tab. The list includes:

- DC-HQ** (9/27/2022, 11:54:55 AM): 6 Devices, 6 Diagnoses. Includes a red error message: 'Destination unreachable from this device' (2) and two successful ping tests for CA-TOR-R1 and CA-TOR-SW1.
- QoS Design - DB Backup Application** (9/14/2022, 12:18:42 AM): 5 Devices, 5 Diagnoses. Includes a green success message: 'The device US-BOS-SW1 's policy map config matches with the baseline' (0) and configuration details for US-BOS-SW1 and US-BOS-R1.
- compound-NI-ospf** (9/26/2022, 9:48:09 AM): 3 Devices, 0 Diagnoses. Includes a configuration for US-BOS-R1 to show ospf neighbors.

The right side of the interface shows a network diagram with various nodes and connections. A red warning icon in the top right corner indicates 'Incident:10001N'.

The user can publish an intent from an NI editor or view mode.



5.1.2 Map Intent

By embedding an intent inside a map (Map Intent), users can have map and intent work side-by-side with an intuitive workflow: create intent along with the map, execute intent while working on the map, and view the intent results in the map.

Users can create a map intent from scratch, associate an existing NI as the map intent, or clone from NIT or NIC.

The screenshot illustrates the workflow for creating a new Map Intent:

- Map Intent Page:** Shows a list of devices and a 'Create' button. A red arrow labeled 'a' points to the 'Select Intent as Template' button.
- Select Intent as Template Dialog:** Displays a search bar and a list of recommended templates. A red arrow labeled 'b' points to the 'OK' button.
- Edit Macro Variables Dialog:** A table for defining variables. A red arrow labeled 'c' points to the 'Continue' button.
- Create Intent Dialog:** Shows the successful creation of an intent named 'map1_01' with a description 'NI HW-4 BGP Timer Miss Match'. A red arrow labeled 'c' points to the 'Save as Map Intent' button.

Hostname	Svrf	Svlan
Bj*POP	red	1
Bj-3750-1		
Bj-3750-2	red	1
Bj-Arista-1	red	10
Bj-Arista-2	red	10
Bj-Avaya-1	red	1
Bj-Avaya-2		

Users can execute the Map Intent, view the diagnosis details, and compare the original data with the data.

The screenshot shows the 'Diagnosis Details and Compare' window for the command 'show ip ospf neighbor' on device 'NBLAB-XR-P1'. It includes a network diagram and a comparison table.

Summary Text	Original Text
2	Internet 3.3.3.3
3	Internet 172.24.10.250
4	Internet 172.24.10.249
5	Internet 172.24.32.225
6	Internet 172.24.32.226
7	Internet 172.24.36.1
8	Internet 172.24.36.2
9	Internet 172.24.10.34
10	Internet 172.24.10.33
11	Internet 199.22.22.7
12	Internet 199.22.22.7
13	Internet 199.21.21.7
14	Internet 199.20.20.7
15	Internet 199.22.22.6
16	Internet 199.21.21.6
17	Internet 199.20.20.6
18	Internet 199.22.22.5
19	Internet 199.21.21.5
20	Internet 199.20.20.5
21	Internet 199.22.22.4
22	Internet 199.21.21.4
23	Internet 199.20.20.4
24	Internet 199.22.22.3
25	Internet 199.21.21.3
26	Internet 199.20.20.3
27	Internet 199.22.22.2
28	Internet 199.21.21.2
29	Internet 199.20.20.2
30	Internet 199.22.22.13
31	Internet 199.21.21.13
32	Internet 199.20.20.13

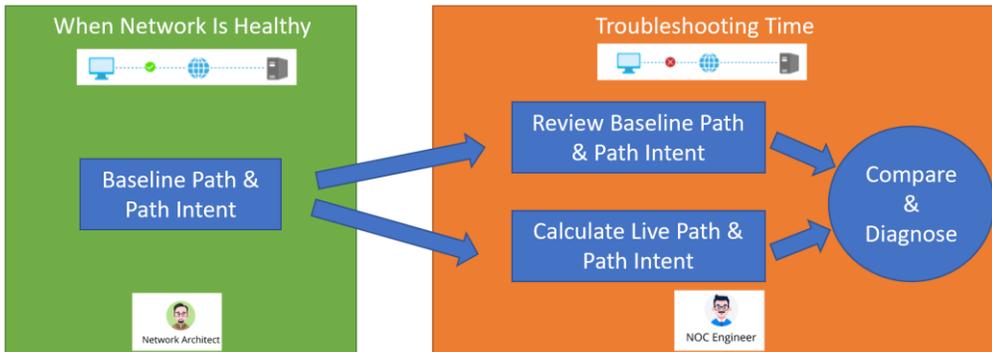
The intent results can also be displayed like a special data view on the map. The devices with alerts will be highlighted and alerted.

The screenshot shows a network management interface with a 'Map Intent' view. The left pane displays diagnostic details for device GW2Lab, including OSPF neighbor and interface status. The right pane shows a network topology map with nodes like GW2Lab, DMVPN, US-LAX-R1, NBLAB_XR-P2, and CA-TOR_R1. Alerts are visible on GW2Lab and US-LAX-R1. A 'Display on Map' button is highlighted over an alert entry.

5.2 Path Intent

The Path-based Troubleshooting Flow (PBTF) is an essential part of PDAS. With PBTF, the network architect can baseline the critical application and path intents when the network is healthy. Path-related baseline data and diagnosis logic can be programmatically defined inside Path Intent, 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. While troubleshooting the application, users

can review the baseline path and path intent results, rerun the associated path NI with the live network data and check the diagnosis results.

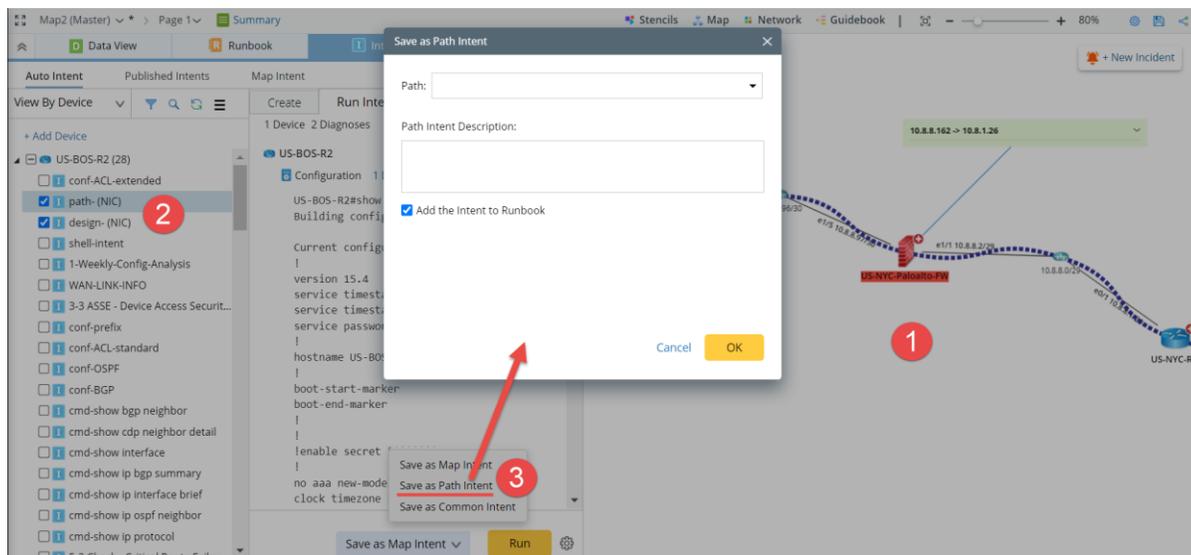


R11 provides more efficient ways to create Path Intent:

- Integrate Auto Intent for Path Intent
- Clone from NIC/NIT Template

5.2.1 Create a Path Intent Via Auto Intent

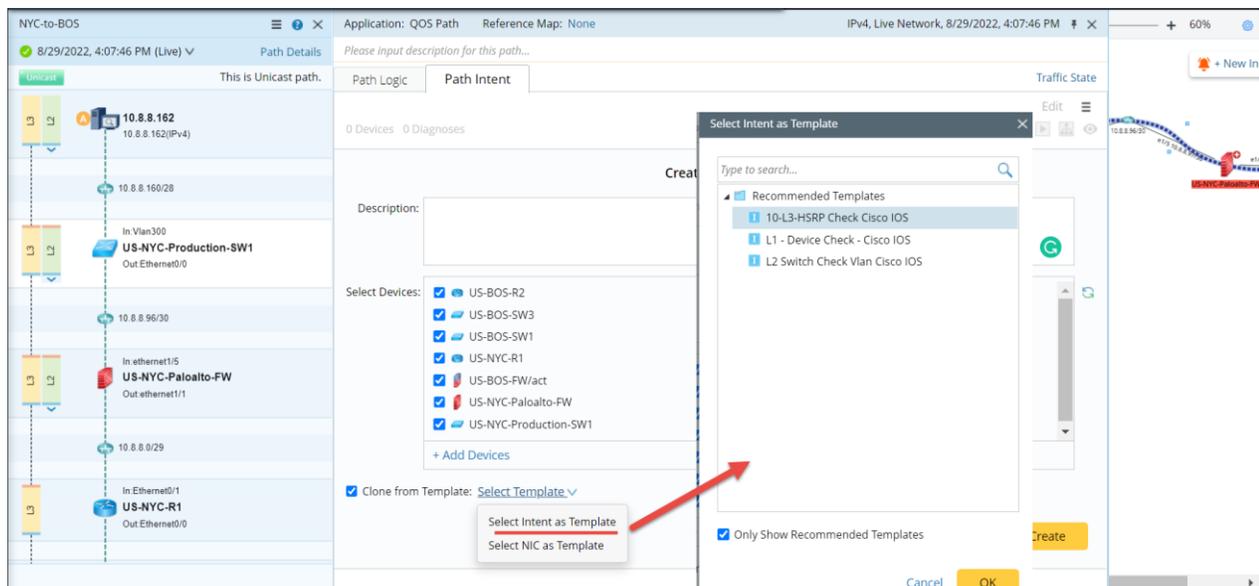
With the Auto Intent feature, users can create the Path Intent with three steps:



1. Map an application path when the network is healthy.
2. Create an intent via Auto Intent.
3. Save the created intent as the path intent.

5.2.2 Create a Path Intent by Cloning from NIC/NIT

Path intent can also be automatically created by using a suitable intent template or cluster.



5.2.3 Run Path NI and View Results

The Path NI pane is improved to support a smooth workflow. Users can

- Execute the Path NI.
- View the diagnosis tree from the Path NI pane.
- Display the Path NI results on the map.

Application: NA 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

Embedded intent for this map. [Edit](#)

3 Devices 10 Diagnosis 06/05/2022 8:55:25 AM

Description: Check OSPF interface cost mismatch.

S Deviation Observed in OSPF Route Summary Table on device GW2Lab ...

BJ_R1 **S** Deviation Observed in OSPF Route Summary Table o...

▶ show ip ospf neighbor 2 diagnosis, 1 alert

10.8.76.206	1	FULL/DR	00:00:30	10.8.76.45...
-------------	---	---------	----------	---------------

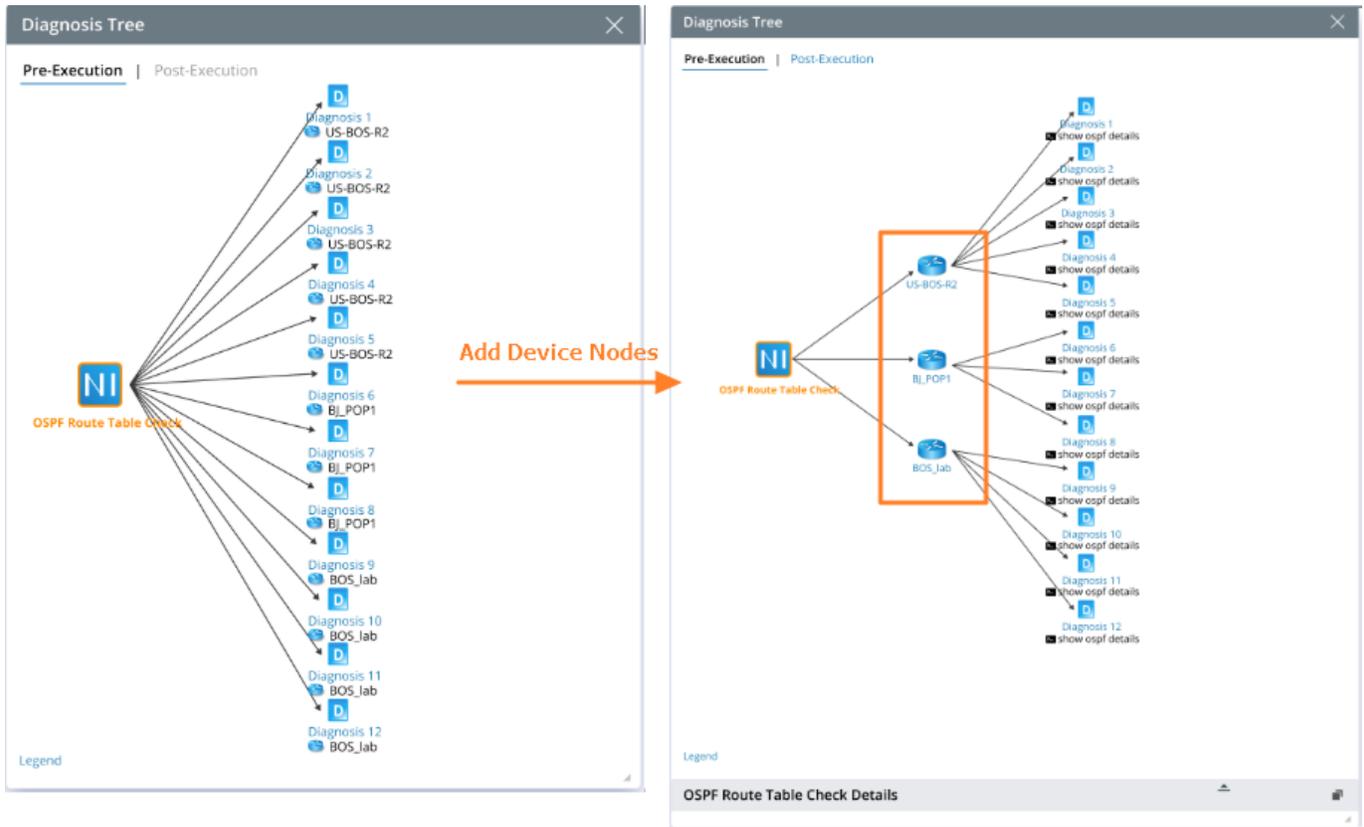
▶ show ip ospf interface

GigabitEthernet0/0/0/0 is up, line protocol is up
Process ID 100, Router ID 10.8.76.46, Network Type BROADCA...

5.3 Diagnosis Tree

R11 improves the diagnosis tree to display more data and add more types of nodes so that the diagnosis tree can be used as a single pane of glass to learn and use NI for diagnosis.

- Add device nodes between the diagnosis node and NI node



In the previous version, when multiple diagnoses are defined under one NI, users are hard to find out the relationship between devices and diagnoses in the diagnosis tree. Adding the device node between the NI node and the diagnosis node optimizes the diagnosis tree layout, providing better readability.

- Provide Detail Pane for All Nodes

A detail pane is provided so the user can view the node details, such as device and diagnosis details.

Diagnosis Tree of 12_check HSRP state and configlet[Cisco IOS]

Source: lihua.zhao111111@netbraintech.com Current NI: 12_check HSRP state and configlet[Cisco IOS] Execution Time: 07/27/2022 10:16:03 AM

Pre-Execution | Post-Execution

Legend

Start → NI → BJ_L2_Core_3

Device Details - BJ_L2_Core_3

Hostname	BJ_L2_Core_3
Mgmt IP	172.24.101.4
Mgmt Interface	Vlan10
Device Type	Cisco IOS Switch
Vendor	Cisco
Model	WS-C3750-24TS
Software Version	12.2(55)SE4
Serial Number	CAT0829Z242
Site	My Network/site_11397_W40
Location	Netbrai-data-valage-40
Contact	Netbrain.com.yipengzhi
System Memory Si...	68217848
Asset Tag	
Hierarchy Layer	
Description	
sysObjectID	1.3.6.1.4.1.9.1.516
Driver	Cisco IOS Switch

Configuration File

```

1 BJ_L2_Core_3#show run
2 Building configuration...
3
4 Current configuration : 10772 bytes
5
6 version 12.2
7 no service pad
8 service timestamps debug datetime msec
9 service timestamps log datetime msec
10 no service password-encryption
11
12 !
13 hostname BJ_L2_Core_3
14 boot-start-marker
15 boot-end-marker
16
17 enable password *****
18
19 !
20 username *****
21 !
22 !
23 aaa new-model
24
25 !
26 aaa authentication login default group tacacs+ local-case
27

```

Diagnosis Tree of 12_check HSRP state and configlet[Cisco IOS]

Source: lihua.zhao111111@netbraintech.com Current NI: 12_check HSRP state and configlet[Cisco IOS] Execution Time: 07/27/2022 10:16:03 AM

Pre-Execution | Post-Execution

Legend

Start → NI → BJ_L2_Core_3

Diagnosis Details - mregex ID line last

Diagnosis Details

Summary Text

Original Text	Summary Text
1 BJ_L2_Core_3#show mls qos interface	
2 FastEthernet1/0/1	FastEthernet1/0/1
3 trust state: not trusted	
4 trust mode: not trusted	
5 trust enabled flag: ena	
6 COS override: dls	
7 default cos: 0	
8 DSCP Mutation Map: Default DSCP Mutation Map	
9 Trust device: none	
10 qos mode: port-based	
11	
12 FastEthernet1/0/2	FastEthernet1/0/2
13 trust state: not trusted	
14 trust mode: not trusted	
15 trust enabled flag: ena	
16 COS override: dls	
17 default cos: 0	
18 DSCP Mutation Map: Default DSCP Mutation Map	
19 Trust device: none	

Diagnosis Logic (mregex ID line last)

Anchor: FastEthernet1/0/1

if

A: BJ_L2_C... Current = BJ_L2_Core_3 Last

\$intf Equals \$intf True

Value: FastEt... Value: FastEt...

Boolean Expression: A True

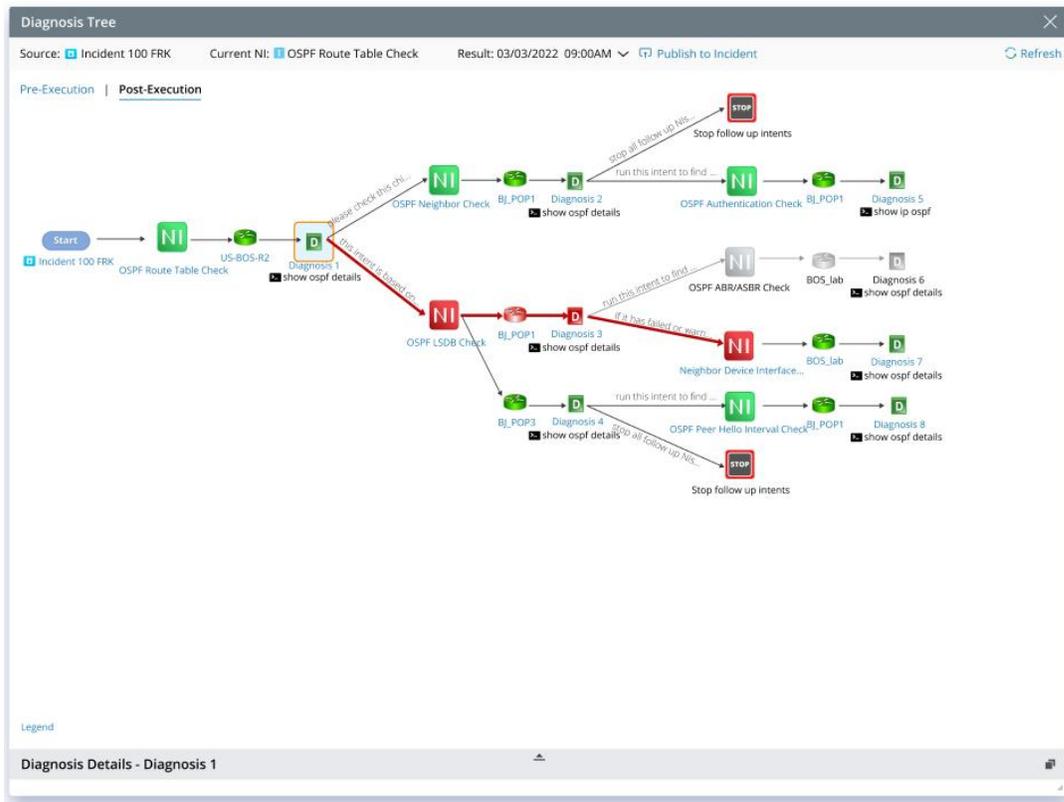
Then

FastEthernet1/0/1

Device Status Code: FastEthernet1/0/1

- Highlight Nodes and Links with Alerts in Red

In the previous version, the diagnosis node in the diagnosis tree does not display the alert status. If an alert is generated, the NI node will turn red, but the diagnosis node will not. Therefore, R11 makes All NI nodes, device nodes, and diagnosis nodes with alerts will turn red.



5.4 Self-Service Tools

PDAS Triggered Automation can be consumed through Machine-to-machine API calls or by a human being through Self-service Tools:

- Incident Portal: each NetBrain incident portal can be enabled as the self-service launcher.

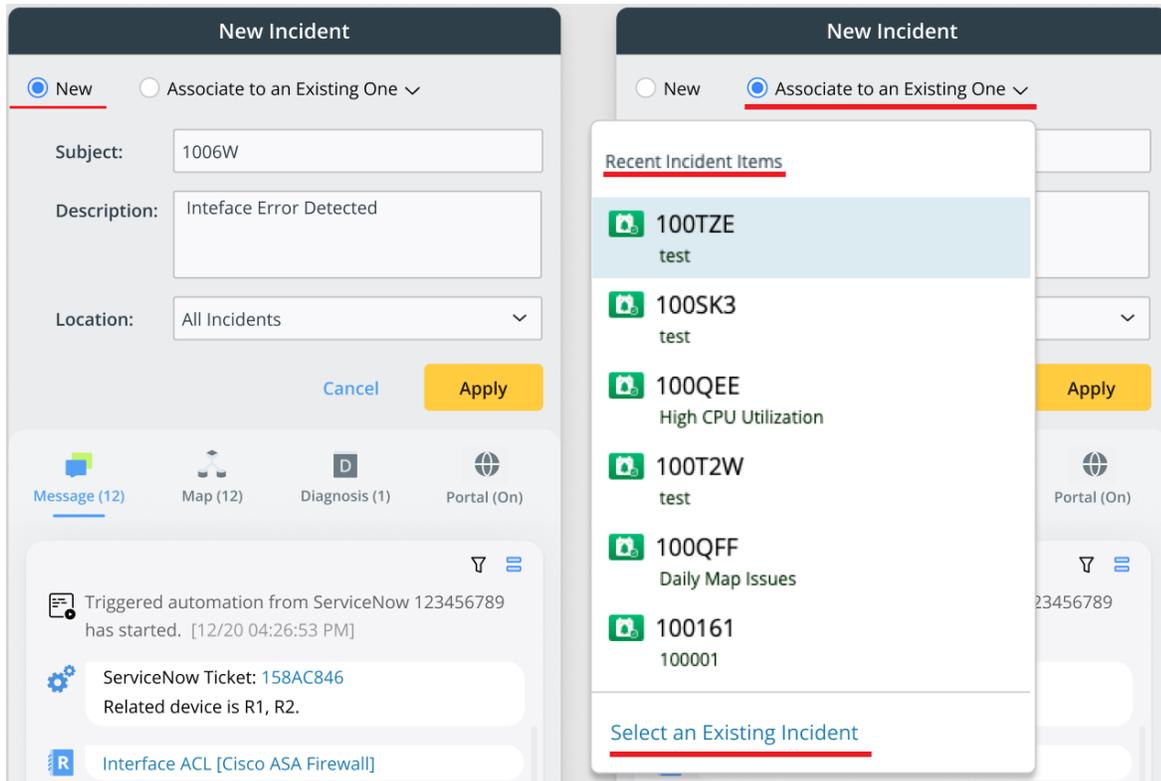
- Email Trigger: fill in an email template and send it to NetBrain to trigger Automation.
- ServiceNow NetBrain App: enable ServiceNow Application to access NetBrain automation.
- Teams Chatbot: deployed through Microsoft Teams store.

Besides improving the Incident Pane/portal, R11 creates a new self-service, **Teams Chatbot**, as a simple way to run NetBrain automation functions and view the results in teams.

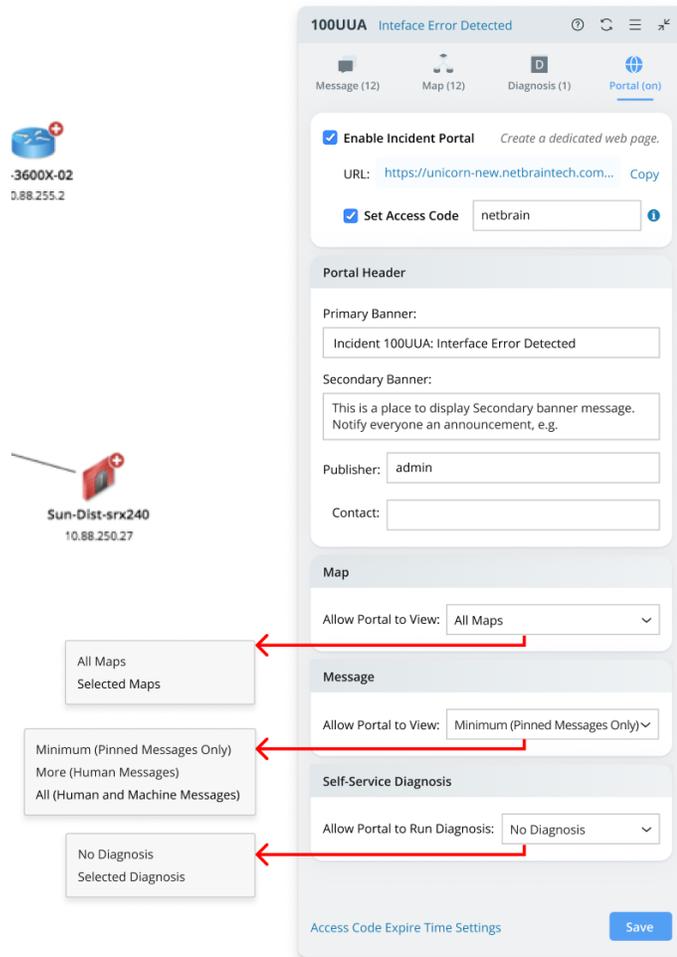
5.4.1 Incident Pane/Portal Enhancements

The Incident feature is further improved to provide End Users with a simple flow of using Incident and Portal:

- Redesign UI to create a new incident: simplify the entrance of creating a new incident and clearly shows two ways to create a new incident: from scratch or associate with an existing one.



- Reserve **Interactive Self-Service Diagnosis Only** and Improve the View of Diagnosis Results. To fully present the interactive automation feature of Incident, the operations performed in the Diagnosis Pane will focus on Self-Service Diagnosis only. The triggered diagnosis results are not shown in the Message pane.
- Emphasize Portal settings: Simplify the flow of setting the incident portal. Users will be more likely to set up and use the portal for smooth collaborative troubleshooting.



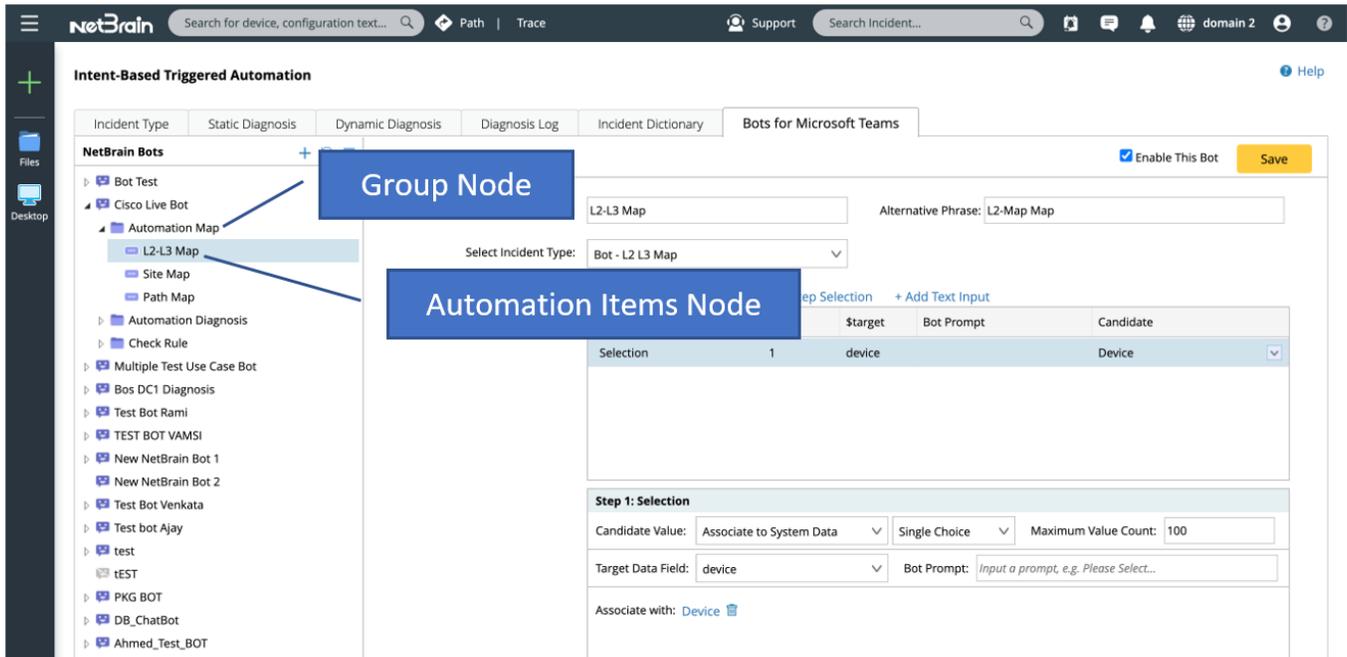
- Share Personal Copy to Incident Portal. Users can also share a personal copy with portal users.

5.4.2 Teams Chatbot

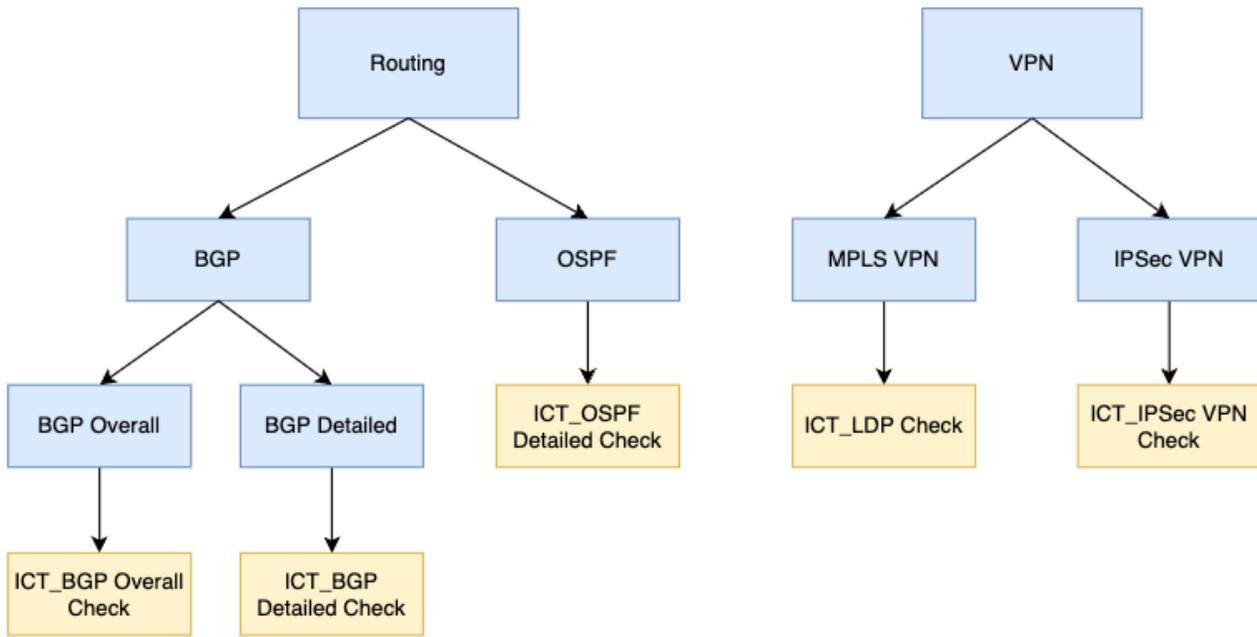
R11 creates a Microsoft Teams Chatbot as a simple way to run NetBrain automation functions and view the results in teams, including:

- Create a network map
- Automate network problem diagnosis
- Enforce design rules and best practices, fully customizable without any coding.

The bot conversation flow can be easily customized in the NetBrain system. The group and automation node form a tree structure to guide the user through the chat.



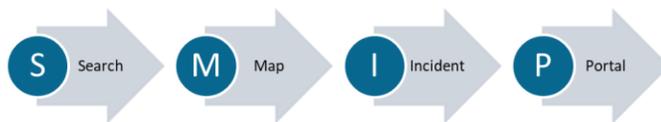
A sample bot for troubleshooting the routing issue:



6 Other Enhancements

6.1 Incident-based Collaboration Workflow Enhancement

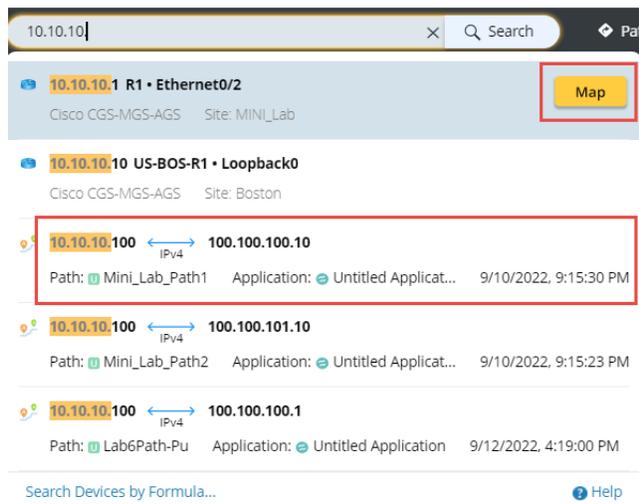
One common interactive collaboration workflow is: Search the digital twin (e.g., IP address), create the map from the search results, open the Incident pane or create a new Incident, and share the incident portal with others.



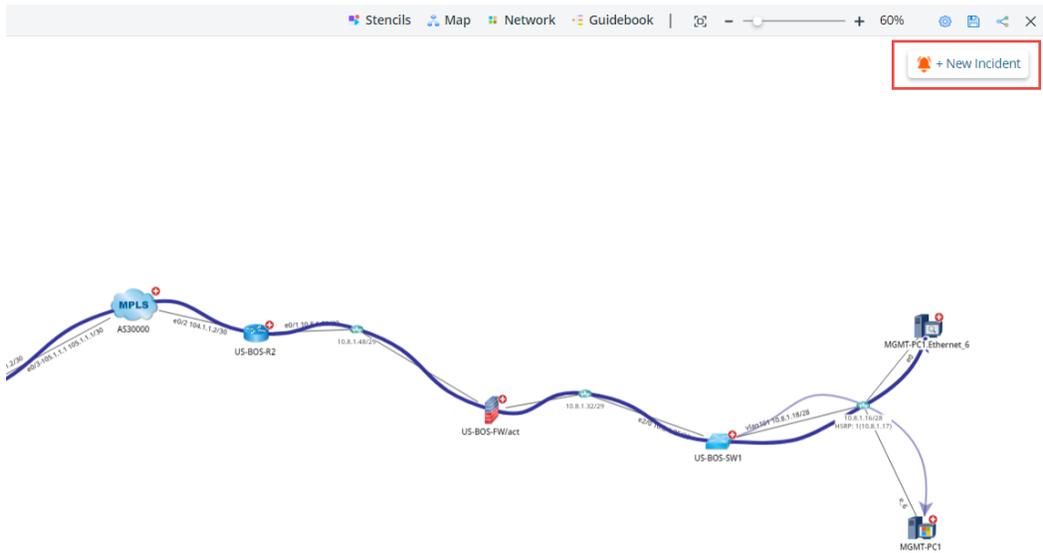
Along this workflow, R11 makes the following improvements:

- Improve the search IntelliSense pane to show more related results and allow users to create the map directly from an entry.

For example, entering an IP address or partial IP address, all device interfaces with IP and historical A/B paths matching the IP address are displayed. You can directly map the interface or the path.



- After mapping the searched result, users can create a new incident from the map, which can be associated with an existing Incident.



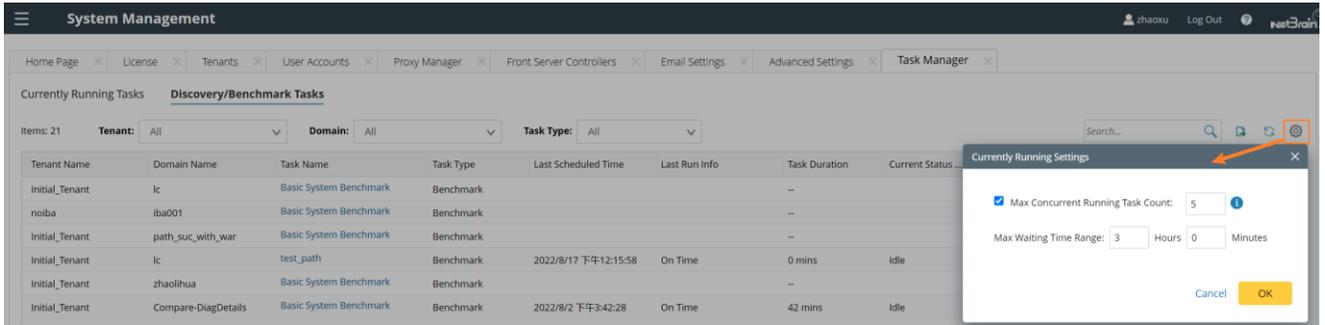
- From the Incident pane, users can enable the Incident Portal to share the map and automation results with others. See [Incident Pane/Portal Enhancements](#) for details.

6.2 Benchmark Schedule Enhancement

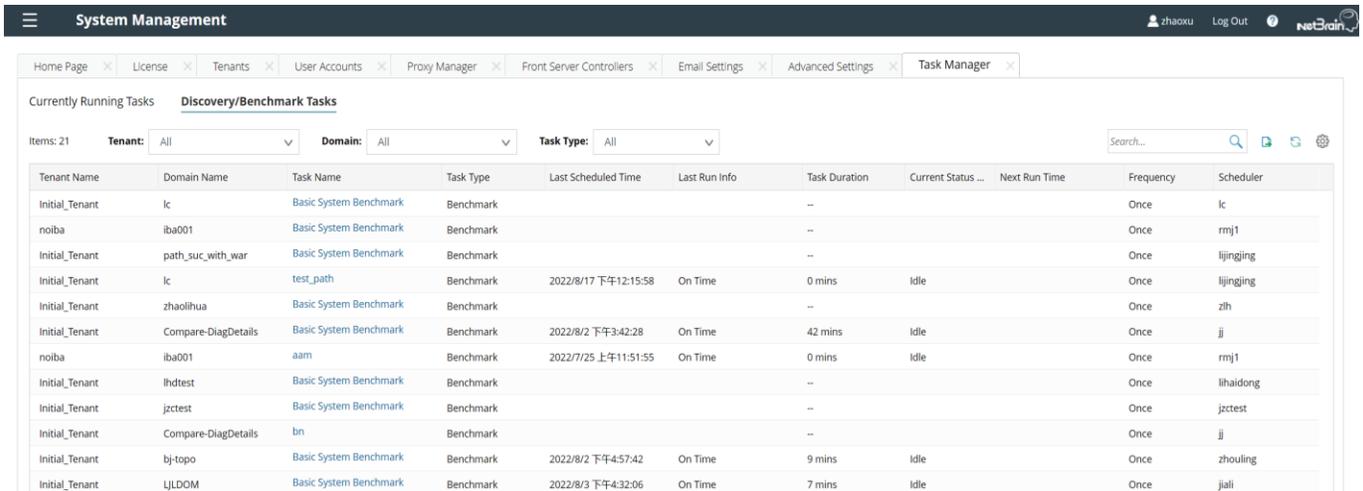
If the system has many domains, such as Service Providers, it is hard to set up the benchmark tasks to limit the concurrent background tasks. NetBrain R11 allows users to configure rules of the concurrently running tasks and view all the scheduled benchmark/discovery tasks on the System Management page to adjust the frequency more easily.

- Configure Rules of Concurrently Running Tasks

Users can set the maximum number of concurrently running tasks to avoid conflicts and specify the maximum waiting time (for running tasks to improve task execution efficiency. The task will be skipped and not executed if the waiting time exceeds the specified waiting time range.



- A **Discovery/Benchmark Tasks** tab is added to the Task Manager on the System Management page to centrally display all the scheduled Discovery/Benchmark tasks across all domains.



- Edit the Task Frequency to Avoid Conflicts

Users can further adjust the task frequency based on the running information to avoid schedule overlapping.

6.3 Email Change Analysis Report

The global change analysis report allows users to track the data change of the entire network and presents the result in the CA Report. In the previous version, if users need to view, save or share a CA report, the only way is to log in to the IE system and manually export the CA report to the local machine. To simplify the process and report network changes via email daily, R11 enables users to receive the CA report via email at a certain frequency.

The scheduled email alert will inform the email recipient of a summary of the CA report, such as the comparison inputs, results, and the involved tenant. Also, a copy of the CA report can be attached to the email as needed.

The screenshot shows the NetBrain Change Analysis Report interface. The main window displays a table of devices and their change details. A modal dialog titled "Email Task Manager" is open, showing a table of email tasks. A red arrow points from the "Email Task Manager" link in the top right of the report to the dialog box.

Change Analysis Report

8/7/2019, 12:00:00 AM | All Devices | All Enabled Data Types | Run Report

Total Changed Devices: 498 out of 1203 Devices | Total Changed Entries: 180397

Device	Data Type	Latest Change Detection Time	Change Details	Total Changes
▶ I@#5%&*0_-=+~:; V[]{} (2)	Row item	Row item	Row item	Row item
▶ .EMU_NAT_R11 (6)				
▶ 11 (4)				
▶ 192 (1)				
▶ 3725 (1)				
▶ 5101_Router (1)				
▶ 64553 (2)				
▶ ACI-ASA-4851 (1)				
▶ ACI-L3OUT-246 (6)				
▶ ACI-L3Out-249 (5)				
▶ ACI-L3OUT-50 (4)				
▶ ACI-L3OUT-50.23 (5)				
▶ ACI-L3OUT50 (5)				
▶ ACI-L3OUT50.22 (5)				
▶ ACI-QA-SW1 (4)				
▶ ACI-SERVICEGRAPH-FW (3)				
▶ ap1200 (2)				
▶ APc471.fe20.4206 (4)				
▶ AS 80001 (1)				
▶ ASA (2)				
▶ ASA,Router (2)				
▶ ASA-Router (3)				

Email Task Manager

Task Type: Change Analysis | Search User...

Enable	Name	Type	Device Scope	Frequency	Creator	To	Cc
<input type="checkbox"/>	Tracey text	Change Analysis	site1	Every 1 weeks on Monday,Tuesday...	zhaoxu (me)	zhaoxu	
<input checked="" type="checkbox"/>	123	Change Analysis	Device group1,...	Every 1 weeks on Monday,Tuesday...		yuheche doctordava mspeidel sjha2	

Close