



# Integrity Check of Candidate Config

This chapter describes how to perform integrity check of Candidate Config.

This chapter includes the following sections:

- [About Candidate Config, on page 1](#)
- [Guidelines and Limitations for Candidate Config Integrity Check, on page 1](#)
- [Performing Integrity Check for Candidate Config, on page 5](#)
- [Examples of Integrity Check, on page 6](#)

## About Candidate Config

Candidate config is a subset of the running-config which checks whether the Candidate config exists in the running-config without any additions or modifications or deletions.

To check the integrity of the candidate config, use the following commands:

- `show diff running-config`
- `show diff startup-config`

For more information on the CLIs, refer to [Performing Integrity Check for Candidate Config, on page 5](#).

## Guidelines and Limitations for Candidate Config Integrity Check

Candidate config integrity check has the following guidelines and limitations:

- Beginning with Cisco NX-OS Release 10.2(3)F, Candidate config integrity check option is introduced on all Cisco Nexus switches.
- If you need to perform an integrity check on full running configuration as input instead of a partial config, then it is recommended not to use "**partial**" keyword.
- The line numbers that are displayed in the generated running config do not match with the candidate config as they are internally generated one.
- If there is any difference between the configuration of running and candidate, then it is displayed inline as output.

- If the whole block of configuration in the candidate file is a new addition, it will be appended at the end of the generated running config.

### Guidelines and Limitations for `show diff running-config file_url [unified] [partial ] [merged] command`

- When using the **unified**, **partial**, and **merged** option to review the differences for the following PBR commands, the diff output are as mentioned below:

- **set ip next-hop**
- **set ip default next-hop**
- **set ip default vrf next-hop**
- **set ipv6 next-hop**
- **set ipv6 default next-hop**
- **set ipv6 default vrf next-hop**

1. If the candidate next-hops are a subset of running next-hops (in same order and sequence), and candidate additive flags are a subset of running flags, then the diff output will be empty as shown in the following table:

Candidate Config	Running Config	Partial Unified Merged Diff Output
route-map rmap1 permit 10 set ip next-hop 1.1.1.1 2.2.2.2 load-share	route-map rmap1 permit 10 set ip next-hop 1.1.1.1 2.2.2.2 3.3.3.3 load-share force-order	<no-diff>

2. If the candidate next-hops are a subset of running next-hops (in same order and sequence), and candidate has some extra additive flags which are not present in running config, then the diff output will append any additional flags present in the candidate config to the running config, similar to command line behavior as shown in the following table:

Candidate Config	Running Config	Partial Unified Merged Diff Output
route-map rmap1 permit 10 set ip next-hop 1.1.1.1 2.2.2.2 load-share force-order	route-map rmap1 permit 10 set ip next-hop 1.1.1.1 2.2.2.2 3.3.3.3 load-share drop-on-fail	route-map rmap1 permit 10 - set ip next-hop 1.1.1.1 2.2.2.2 3.3.3.3 load-share drop-on-fail + set ip next-hop 1.1.1.1 2.2.2.2 3.3.3.3 load-share force-order drop-on-fail

3. If candidate next-hops are not a subset of running next-hops (in same order and sequence), and the candidate and running record can have any additive flag, then the diff output will indicate this with a '-' for the running config record and a '+' for the candidate config record.

This distinction is important, particularly when using with PBR commands, where the sequence of next-hops is critical. Even if the next-hops IP addresses are identical, their order affects functionality.

For example, '1.1.1.1 2.2.2.2' is different from '2.2.2.2 1.1.1.1'.

**Important**

If there is an additive flag in the running config that you wish to retain after merging with the candidate config, you must explicitly include that flag in the candidate config. This ensures that the desired flags are preserved in the final, merged configuration.

Candidate Config	Running Config	Partial Unified Merged Diff Output
<pre>route-map rmap1 permit 10  set ip next-hop 1.1.1.1  2.2.2.2 load-share  drop-on-fail</pre>	<pre>route-map rmap1 permit 10  set ip next-hop 2.2.2.2  1.1.1.1 load-share  force-order</pre>	<pre>route-map rmap1 permit 10 - set ip next-hop 2.2.2.2  1.1.1.1 load-share  force-order + set ip next-hop 1.1.1.1  2.2.2.2 load-share  drop-on-fail</pre>

- When **Partial Unified** or **Partial Unified Merged** option is used, all the PBR commands are mutually exclusive and cannot coexist within the same parent route-map. Therefore, if a candidate configuration specifies multiple mutually exclusive PBR commands under a single route-map, only the last command variant will be shown in the partial diff output.

Example-1: In this example, the candidate configuration includes multiple PBR commands under a single route-map **rmap1**:

```
route-map rmap1 permit 10
 set ip next-hop 1.1.1.1 2.2.2.2
 set ipv6 next-hop 3::3
 set ip next-hop verify-availability 4.4.4.4
 set ip next-hop verify-availability 5.5.5.5
 set ip vrf green next-hop 6.6.6.6
 set ip vrf blue next-hop 7.7.7.7 8.8.8.8
```

Prior to the generation of the partial-diff output, the above candidate configuration will be automatically converted to the following:

```
route-map rmap1 permit 10
 set ip vrf green next-hop 6.6.6.6
 set ip vrf blue next-hop 7.7.7.7 8.8.8.8
```

Example-2: In this example, the candidate configuration includes multiple 'set ip next-hop verify-availability' commands with different track IDs specified for the route-map **rmap2**. Since track IDs cannot be modified for the same next-hop, these commands are mutually exclusive:

```
route-map rmap2 permit 10
 set ip next-hop verify-availability 1.1.1.1 track 1
 set ip next-hop verify-availability 2.2.2.2 track 20
 set ip next-hop verify-availability 2.2.2.2 track 30
 set ip next-hop verify-availability 2.2.2.2 track 40
 set ip next-hop verify-availability 3.3.3.3 track 3
```

Before generating the partial-diff output, the system will automatically consolidate these commands by retaining only the last **set ip next-hop verify-availability** command for each next-hop IP address as shown below:

```
route-map rmap2 permit 10
 set ip next-hop verify-availability 1.1.1.1 track 1
 set ip next-hop verify-availability 2.2.2.2 track 40
 set ip next-hop verify-availability 3.3.3.3 track 3
```

- When **Partial Unified Merged** option is used, to review the differences for the **verify-availability** command variants, the track ID for a given next-hop is not modifiable.

Consequently, if the candidate and running configurations contain the same next-hop but have different track IDs under the same parent route-map, the candidate record cannot simply be merged with the running record, as per command line behavior. Therefore, in order to apply the candidate record with different track ID for the same next-hop, the corresponding running config record must be removed first ('-' for the running configuration record in the diff) and then when the candidate record is merged, it will be appended at the end of the last record under the same parent route-map ('+' for the candidate config record).

The following table shows the sample candidate and running configuration with the **Partial Unified Merged** output for different use cases as mentioned below:

- If track ID is different for the same next-hop under candidate and running config, then the diff output will be as shown in the following table:

Candidate Config	Running Config	Partial Unified Merged Diff Output
<pre>route-map rmap1 permit 10   set ip next-hop   verify-availability 1.1.1.1   track 1   set ip next-hop   verify-availability 2.2.2.2   track 20   set ip next-hop   verify-availability 3.3.3.3   track 3 load-share</pre>	<pre>route-map rmap1 permit 10   set ip next-hop   verify-availability 1.1.1.1   track 1   set ip next-hop   verify-availability 2.2.2.2   track 2   set ip next-hop   verify-availability 3.3.3.3   track 3 load-share</pre>	<pre>route-map test permit 10   set ip next-hop   verify-availability 1.1.1.1   track 1   - set ip next-hop   verify-availability 2.2.2.2   track 2   set ip next-hop   verify-availability 3.3.3.3   track 3   + set ip next-hop   verify-availability 2.2.2.2   track 20 load-share</pre>

- If track ID is not present under candidate config but present in running config for the same next-hop, then the diff output will be empty as shown in the following table:

Candidate Config	Running Config	Partial Unified Merged Diff Output
<pre>route-map rmap1 permit 10   set ip next-hop   verify-availability 1.1.1.1   track 1   set ip next-hop   verify-availability 2.2.2.2   track 2   set ip next-hop   verify-availability 3.3.3.3   track 3</pre>	<pre>route-map rmap1 permit 10   set ip next-hop   verify-availability 1.1.1.1   track 1   set ip next-hop   verify-availability 2.2.2.2   track 2   set ip next-hop   verify-availability 3.3.3.3   track 3</pre>	<no-diff>

- If track ID is not present under running config but present in candidate config for the same next-hop, then the diff output will be as shown in the following table:

Candidate Config	Running Config	Partial Unified Merged Diff Output
<pre>route-map rmap1 permit 10   set ip next-hop verify-availability 1.1.1.1 track 1   set ip next-hop verify-availability 2.2.2.2 track 20   set ip next-hop verify-availability 3.3.3.3 track 3</pre>	<pre>route-map rmap1 permit 10   set ip next-hop verify-availability 1.1.1.1 track 1   set ip next-hop verify-availability 2.2.2.2   set ip next-hop verify-availability 3.3.3.3 track 3</pre>	<pre>route-map rmap1 permit 10   set ip next-hop verify-availability 1.1.1.1 track 1 - set ip next-hop verify-availability 2.2.2.2   set ip next-hop verify-availability 3.3.3.3 track 3 + set ip next-hop verify-availability 2.2.2.2 track 20</pre>

## Performing Integrity Check for Candidate Config

To perform the integrity check, use the following commands:

### Before you begin



**Note** Before performing the integrity check, ensure that the running config and the candidate config belong to the same image version.

### Procedure

	Command or Action	Purpose
<b>Step 1</b>	<pre>show diff running-config <i>file_url</i> [unified] [partial]</pre> <p><b>Example:</b></p> <pre>switch# show diff running-config bootflash:candidate.cfg partial unified</pre>	<p>Displays the differences between the running and user given candidate config.</p> <ul style="list-style-type: none"> <li>• <i>file_url</i>: File path to compare with.</li> <li>• <b>unified</b>: Displays the difference between running and user configuration in unified format.</li> <li>• <b>partial</b>: Enter <b>partial</b> only if user configuration file is partial and not a full configuration.</li> </ul>
<b>Step 2</b>	<pre>show diff startup-config <i>file_url</i> [unified]</pre> <p><b>Example:</b></p> <pre>switch# show diff startup-config bootflash:candidate.cfg unified</pre>	<p>Displays the differences between the startup and user given candidate config.</p> <ul style="list-style-type: none"> <li>• <i>file_url</i>: File path to compare with.</li> <li>• <b>unified</b>: Displays the difference between startup and user configuration in unified format.</li> </ul>

# Examples of Integrity Check

## No Difference Between Running and Candidate Config

```
switch# show diff running-config bootflash:base_running.cfg
switch#
```

## Difference Between Running and Candidate

```
switch# show diff running-config bootflash:modified-running.cfg unified
--- running-config
+++ User-config
@@ -32,11 +32,11 @@

interface Ethernet1/1
    mtu 9100
    link debounce time 0
    beacon
- ip address 2.2.2.2/24
+ ip address 1.1.1.1/24
    no shutdown

interface Ethernet1/2

interface Ethernet1/3
switch#
```

## Difference Between Running and Partial Candidate

```
switch# show file bootflash:intf_vlan.cfg
interface Vlan101
    no shutdown
    no ip redirects
    ip address 1.1.2.1/24 secondary
    ip address 1.1.1.1/24
switch#
switch# show diff running-config bootflash:intf_vlan.cfg partial unified
--- running-config
+++ User-config
@@ -3897,10 +3883,14 @@
    mtu 9100
    ip access-group IPV4_EDGE in
    ip address 2.2.2.12/26 tag 54321

interface Vlan101
+ no shutdown
+ no ip redirects
+ ip address 1.1.2.1/24 secondary
+ ip address 1.1.1.1/24

interface Vlan102
    description Vlan102
    no shutdown
    mtu 9100
switch#
```