# Configuring Object Tracking

This chapter describes how to configure object tracking on Cisco NX-OS switches.

This chapter includes the following sections:


- Information About Object Tracking, on page 1
- Guidelines and Limitations for Object Tracking, on page 2
- Default Settings for Object Tracking, on page 3
- Configuring Object Tracking, on page 3
- Verifying the Object Tracking Configuration, on page 12
- Configuration Examples for Object Tracking, on page 13
- Related Topics, on page 13
- Additional References, on page 13


# Information About Object Tracking

Object tracking allows you to track specific objects on the switch, such as the interface line protocol state, IP routing, and route reachability, and to take action when the tracked object's state changes. This feature allows you to increase the availability of the network and shorten recovery time if an object state goes down.

## Object Tracking Overview

The object tracking feature allows you to create a tracked object that multiple clients can use to modify the client behavior when a tracked object changes. Several clients register their interest with the tracking process, track the same object, and take different actions when the object state changes.

Clients include the following features:

- Hot Standby Redundancy Protocol (HSRP)
- Virtual Router Redundancy Protocol (VRRP)
- Embedded Event Manager (EEM)

The object tracking monitors the status of the tracked objects and communicates any changes made to interested clients. Each tracked object is identified by a unique number that clients can use to configure the action to take when a tracked object changes state.

Cisco NX-OS tracks the following object types:

- Interface line protocol state—Tracks whether the line protocol state is up or down.
- Interface IP routing state—Tracks whether the interface has an IPv4 address and if IPv4 routing is enabled and active.
- IP route reachability—Tracks whether an Ipv4 route exists and is reachable from the local switch.

For example, you can configure HSRP to track the line protocol of the interface that connects one of the redundant routers to the rest of the network. If that link protocol goes down, you can modify the priority of the affected HSRP router.

## Object Track List

An object track list allows you to track the combined states of multiple objects. Object track lists support the following capabilities:

- Boolean "and" function—Each object defined within the track list must be in an up state so that the track list object can become up.
- Boolean "or" function—At least one object defined within the track list must be in an up state so that the tracked object can become up.
- Threshold percentage—The percentage of up objects in the tracked list must be greater than the configured up threshold for the tracked list to be in the up state. If the percentage of down objects in the tracked list is above the configured track list down threshold, the tracked list is marked as down.
- Threshold weight—Assign a weight value to each object in the tracked list, and a weight threshold for the track list. If the combined weights of all up objects exceeds the track list weight up threshold, the track list is in an up state. If the combined weights of all the down objects exceeds the track list weight down threshold, the track list is in the down state.

See the Configuring an Object Track List with a Boolean Expression section for more information on track lists.

## Virtualization Support

Object tracking supports Virtual Routing and Forwarding (VRF) instances. By default, Cisco NX-OS places you in the default VRF unless you specifically configure another VRF. By default, Cisco NX-OS tracks the route reachability state of objects in the default VRF. If you want to track objects in another VRF, you must configure the object to be a member of that VRF (see the Configuring Object Tracking for a Nondefault VRF section).

For more information, see Configuring Layer 3 Virtualization.

# Guidelines and Limitations for Object Tracking

Object tracking has the following configuration guidelines and limitations:

- Supports up to 500 tracked objects.
- Supports Ethernet, subinterfaces, tunnels, port channels, loopback interfaces, and VLAN interfaces.
- Supports one tracked object per HSRP group.

# Default Settings for Object Tracking

Table below lists the default settings for object tracking parameters.

***Table 1: Default Object Tracking Parameters***

| Parameters | Default |
|---|---|
| Tracked Object VRF | Member of default VRF |

# Configuring Object Tracking

## Configuring Object Tracking for an Interface

You can configure Cisco NX-OS to track the line protocol or IPv4 routing state of an interface.

### SUMMARY STEPS

1. **configure terminal**
2. **track** *object-id* **interface** *interface-type number* { **ip routing** | **line-protocol** }
3. (Optional) **show track** [ *object-id* ]
4. (Optional) **copy running-config startup-config**

### DETAILED STEPS

#### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **track** *object-id* **interface** *interface-type number* { **ip routing** | **line-protocol** }<br>**Example:**<br>`switch(config)# track 1 interface ethernet 1/2 line-protocol`<br>`switch(config-track#` | Creates a tracked object for an interface and enters tracking configuration mode. The *object-id* range is from 1 to 500. |
| **Step 3** | (Optional) **show track** [ *object-id* ]<br>**Example:**<br>`switch(config-track)# show track 1` | Displays object tracking information |

| | Command or Action | Purpose |
|---|---|---|
| **Step 4** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch(config-track)# copy running-config startup-config` | Saves this configuration change. |

### Example

This example shows how to configure object tracking for the line protocol state on Ethernet 1/2:

```
switch# configure terminal
switch(config)# track 1 interface ethernet 1/2 line-protocol
switch(config-track)# copy running-config startup-config
```

This example shows how to configure object tracking for the IPv4 routing state on Ethernet 1/2:

```
switch# configure terminal
switch(config)# track 2 interface ethernet 1/2 ip routing
switch(config-track)# copy running-config startup-config
```

# Configuring Object Tracking for Route Reachability

You can configure Cisco NX-OS to track the existence and reachability of an IP route.

### SUMMARY STEPS

1. **configure terminal**
2. **track** *object-id* **ip route** *prefix/length* **reachability**
3. (Optional) **show track** [ *object-id* ]
4. (Optional) **copy running-config startup-config**

### DETAILED STEPS

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **track** *object-id* **ip route** *prefix/length* **reachability**<br>**Example:**<br>`switch(config)# track 2 ip route 192.0.2.0/8 reachability`<br>`switch(config-track)#` | Creates a tracked object for a route and enters tracking configuration mode. The *object-id* range is from 1 to 500. The prefix format for IP is A.B.C.D/length, where the length range is from 1 to 32. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | (Optional) **show track** [ *object-id* ]<br>**Example:**<br>`switch(config-track)# show track 1` | Displays object tracking information |
| **Step 4** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch(config-track)# copy running-config startup-config` | Saves this configuration change. |

### Example

This example shows how to configure object tracking for an IPv4 route in the default VRF:

```
switch# configure terminal
switch(config)# track 4 ip route 192.0.2.0/8 reachability
switch(config-track)# copy running-config startup-config
```

# Configuring an Object Track List with a Boolean Expression

You can configure an object track list that contains multiple tracked objects. A tracked list contains one or more objects. The Boolean expression enables two types of calculation by using either "and" or "or" operators. For example, when tracking two interfaces using the "and" operator, up means that both interfaces are up, and down means that either interface is down.

### SUMMARY STEPS

1. **configure terminal**
2. **track** *track-number* **list boolean** { **and** | **or** }
3. **object** *object-id* [ **not** ]
4. (Optional) **show track**
5. (Optional) **copy running-config startup-config**

### DETAILED STEPS

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **track** *track-number* **list boolean** { **and** | **or** }<br>**Example:** | Configures a tracked list object and enters tracking configuration mode. Specifies that the state of the tracked |

| | Command or Action | Purpose |
|---|---|---|
| | `switch(config)# track 1 list boolean and`<br>`switch(config-track#` | list is based on a Boolean calculation. The keywords are as follows:<br>• and —Specifies that the list is up if all objects are up, or down if one or more objects are down. For example, when tracking two interfaces, up means that both interfaces are up, and down means that either interface is down.<br>• **or** —Specifies that the list is up if at least one object is up. For example, when tracking two interfaces, up means that either interface is up, and down means that both interfaces are down.<br>The *track-number* range is from 1 to 500. |
| **Step 3** | **object** *object-id* [ **not** ]<br>**Example:**<br>`switch(config-track)# object 10` | Adds a tracked object to the track list. The *object-id* range is from 1 to 500. The **not** keyword optionally negates the tracked object state.<br>**Note**<br>The example means that when object 10 is up, the tracked list detects object 10 as down. |
| **Step 4** | (Optional) **show track**<br>**Example:**<br>`switch(config-track)# show track` | Displays object tracking information |
| **Step 5** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch(config-track)# copy running-config startup-config` | Saves this configuration change. |

### Example

This example shows how to configure a track list with multiple objects as a Boolean “and”:

```
switch# configure terminal
switch(config)# track 1 list boolean and
switch(config-track)# object 10
switch(config-track)# object 20 not
```

# Configuring an Object Track List with a Percentage Threshold

You can configure an object track list that contains a percentage threshold. A tracked list contains one or more objects. The percentage of up objects must exceed the configured track list up percent threshold before the track list is in an up state. For example, if the tracked list has three objects, and you configure an up threshold of 60 percent, two of the objects must be in the up state (66 percent of all objects) for the track list to be in the up state.

## SUMMARY STEPS

1. **configure terminal**
2. **track** *track-number* **list threshold percentage**
3. **threshold percentage up** *up-value* **down** *down-value*
4. (Optional) **object** [ *object-id* ]
5. (Optional) **show track**
6. (Optional) **copy running-config startup-config**

## DETAILED STEPS

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **track** *track-number* **list threshold percentage**<br>**Example:**<br>`switch(config)# track 1 list threshold percentage`<br>`switch(config-track#` | Configures a tracked list object and enters tracking configuration mode. Specifies that the state of the tracked list is based on a configured threshold percent.<br>The *track-number* range is from 1 to 500. |
| **Step 3** | **threshold percentage up** *up-value* **down** *down-value*<br>**Example:**<br>`switch(config-track)# threshold percentage up 70 down 30` | Configures the threshold percent for the tracked list. The range from 0 to 100 percent. |
| **Step 4** | (Optional) **object** [ *object-id* ]<br>**Example:**<br>`switch(config-track)# object 10` | Adds a tracked object to the track list. The *object-id* range is from 1 to 500. |
| **Step 5** | (Optional) **show track**<br>**Example:**<br>`switch(config-track)# show track` | Displays object tracking information |
| **Step 6** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch(config-track)# copy running-config startup-config` | Saves this configuration change. |

### Example

This example shows how to configure a track list with an up threshold of 70 percent and a down threshold of 30 percent:

```
switch# configure terminal
switch(config)# track 1 list threshold percentage
switch(config-track)# threshold percentage up 70 down 30
switch(config-track)# object 10
switch(config-track)# object 20
switch(config-track)# object 30
```

# Configuring an Object Track List with a Weight Threshold

You can configure an object track list that contains a weight threshold. A tracked list contains one or more objects. The combined weight of up objects must exceed the configured track list up weight threshold before the track list is in an up state. For example, if the tracked list has three objects with the default weight of 10 each, and you configure an up threshold of 15, two of the objects must be in the up state (combined weight of 20) for the track list to be in the up state.

### SUMMARY STEPS

1. **configure terminal**
2. **track** *track-number* **list threshold weight**
3. **threshold weight up** *up-value* **down** *down-value*
4. **object** *object-id* **weight** *value*
5. (Optional) **show track**
6. (Optional) **copy running-config startup-config**

### DETAILED STEPS

#### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | **track** *track-number* **list threshold weight**<br>**Example:**<br>`switch(config)# track 1 list threshold weight`<br>`switch(config-track#` | Configures a tracked list object and enters tracking configuration mode. Specifies that the state of the tracked list is based on a configured threshold weight.<br>The *track-number* range is from 1 to 500. |
| **Step 3** | **threshold weight up** *up-value* **down** *down-value*<br>**Example:**<br>`switch(config-track)# threshold weight up 30 down 10` | Configures the threshold weight for the tracked list. The range from 1 to 255. |
| **Step 4** | **object** *object-id* **weight** *value*<br>**Example:**<br>`switch(config-track)# object 10 weight 15` | Adds a tracked object to the track list. The *object-id* range is from 1 to 500. The *value* range is from 1 to 255. The default weight value is 10. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 5** | (Optional) **show track**<br>**Example:**<br>`switch(config-track)# show track` | Displays object tracking information |
| **Step 6** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch(config-track)# copy running-config startup-config` | Saves this configuration change. |

### Example

This example shows how to configure a track list with an up weight threshold of 30 and a down threshold of 10:

```
switch# configure terminal
switch(config)# track 1 list threshold weight
switch(config-track)# threshold weight up 30 down 10
switch(config-track)# object 10 weight 15
switch(config-track)# object 20 weight 15
switch(config-track)# object 30
```

In this example, the track list is up if object 10 and object 20 are up, and the track list goes to the down state if all three objects are down.

# Configuring an Object Tracking Delay

You can configure a delay for a tracked object or an object track list that delays when the object or list triggers a stage change. The tracked object or track list starts the delay timer when a state change occurs but does not recognize a state change until the delay timer expires. At that point, Cisco NX-OS checks the object state again and records a state change only if the object or list currently has a changed state. Object tracking ignores any intermediate state changes before the delay timer expires.

For example, for an interface line-protocol tracked object that is in the up state with a 20-second down delay, the delay timer starts when the line protocol goes down. The object is not in the down state unless the line protocol is down 20 seconds later.

You can configure independent up delay and down delay for a tracked object or track list. When you delete the delay, object tracking deletes both the up and down delay.

You can change the delay at any point. If the object or list is already counting down the delay timer from a triggered event, the new delay is computed as the following:

- If the new configuration value is less than the old configuration value, the timer starts with the new value .
- If the new configuration value is more than the old configuration value, the timer is calculated as the new configuration value minus the current timer countdown minus the old configuration value.

### SUMMARY STEPS

1. **configure terminal**
2. track *object- id* { *parameters* }
3. **track** *track-number* **list**{ *parameters* }
4. **delay** { **up** *up-time* [**down** *down-time* ] | **down** *down-time* [ **up** *up-time*]}
5. (Optional) **show track**
6. (Optional) **copy running-config startup-config**

### DETAILED STEPS

#### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`switch# configure terminal`<br>`switch(config)#` | Enters configuration mode. |
| **Step 2** | track *object- id* { *parameters* }<br>**Example:**<br>`switch(config)# track 2 ip route 192.0.2.0/8 reachability`<br>`switch(config-track)#` | Creates a tracked object for a route and enters tracking configuration mode. The *object-id* range is from 1 to 500. The prefix format for IP is A.B.C.D/length, where the length range is from 1 to 32. |
| **Step 3** | **track** *track-number* **list**{ *parameters* }<br>**Example:**<br>`switch(config)# track 1 list threshold weight`<br>`switch(config-track#` | Configures a tracked list object and enters tracking configuration mode. Specifies that the state of the tracked list is based on a configured threshold weight.<br>The *track-number* range is from 1 to 500. |
| **Step 4** | **delay** { **up** *up-time* [**down** *down-time* ] | **down** *down-time* [ **up** *up-time*]}<br>**Example:**<br>`switch(config-track)# delay up 20 down 30` | Configures the object delay timers. The range is from 0 to 180 seconds. |
| **Step 5** | (Optional) **show track**<br>**Example:**<br>`switch(config-track)# show track` | Displays object tracking information |
| **Step 6** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch(config-track)# copy running-config startup-config` | Saves this configuration change. |

#### Example

This example shows how to configure object tracking for a route and use delay timers:

```
switch# configure terminal
switch(config)# track 2 ip route 209.165.201.0/8 reachability
switch(config-track)# delay up 20 down 30
switch(config-track)# copy running-config startup-config
```

This example shows how to configure a track list with an up weight threshold of 30 and a down threshold of 10 with delay timers:

```
switch# configure terminal
switch(config)# track 1 list threshold weight
switch(config-track)# threshold weight up 30 down 10
switch(config-track)# object 10 weight 15
switch(config-track)# object 20 weight 15
switch(config-track)# object 30
switch(config-track)# delay up 20 down 30
```

This example shows the delay timer in the show track command output before and after an interface is shut down:

```
switch(config-track)# show track
Track 1
Interface loopback1 Line Protocol
Line Protocol is UP
1 changes, last change 00:00:13
Delay down 10 secs

switch(config-track)# interface loopback 1
switch(config-if)# shutdown
switch(config-if)# show track
Track 1
Interface loopback1 Line Protocol
Line Protocol is delayed DOWN (8 secs remaining)<------- delay timer counting down
1 changes, last change 00:00:22
Delay down 10 secs
```

# Configuring Object Tracking for a Nondefault VRF

You can configure Cisco NX-OS to track an object in a specific VRF.

### SUMMARY STEPS

1. **configure terminal**
2. **track object-id ip route** *prefix/length* **reachability**
3. **vrf member** *vrf-name*
4. (Optional) **show track** [ *object-id* ]
5. (Optional) **copy running-config startup-config**

### DETAILED STEPS

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:** | Enters configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| | `switch# configure terminal`<br>`switch(config)#` | |
| **Step 2** | **track object-id ip route** *prefix/length* **reachability**<br>**Example:**<br>`switch(config)# track 2 ip route 192.0.2.0/8 reachability`<br>`switch(config-track)#` | Creates a tracked object for a route and enters tracking configuration mode. The *object-id* range is from 1 to 500. The prefix format for IP is A.B.C.D/length, where the length range is from 1 to 32. |
| **Step 3** | **vrf member** *vrf-name*<br>**Example:**<br>`switch(config-track)# vrf member Red` | Configures the VRF to use for tracking the configured object. |
| **Step 4** | (Optional) **show track** [ *object-id* ]<br>**Example:**<br>`switch(config-track)# show track 3` | Displays object tracking information |
| **Step 5** | (Optional) **copy running-config startup-config**<br>**Example:**<br>`switch(config-track)# copy running-config startup-config` | Saves this configuration change. |

#### Example

This example shows how to configure object tracking for a route and use VRF Red to look up reachability information for this object:

```
switch# configure terminal
switch(config)# track 2 ip route 209.165.201.0/8 reachability
switch(config-track)# vrf member Red
switch(config-track)# copy running-config startup-config
```

This example shows how to modify tracked object 2 to use VRF Blue instead of VRF Red to look up reachability information for this object:

```
switch# configure terminal
switch(config)# track 2
switch(config-track)# vrf member Blue
switch(config-track)# copy running-config startup-config
```

# Verifying the Object Tracking Configuration

To display the object tracking configuration information, perform one of the following tasks:

| Command | Purpose |
|---|---|
| **show track** [ *object-id* ] [ **brief** ] | Displays the object tracking information for one or more objects. |
| **show track** [ *object-id* ] **interface** [**brief** ] | Displays the interface-based object tracking information. |

| Command | Purpose |
|---|---|
| **show track** [ *object-id* ] **ip-route** [**brief** ] | Displays the IPv4 route-based object tracking information. |

# Configuration Examples for Object Tracking

This example shows how to configure object tracking for route reachability and use VRF Red to look up reachability information for this route:

```
switch# configure terminal
switch(config)# track 2 ip route 209.165.201.0/8 reachability
switch(config-track)# vrf member Red
switch(config-track)# copy running-config startup-config
```

# Related Topics

See the following topics for information related to object tracking:

- Configuring Layer 3 Virtualization
- Configuring HSRP

# Additional References

For additional information related to implementing object tracking, see the following sections:

## Related Documents

| Related Topic | Document Title |
|---|---|
| Configuring the Embedded Event Manager | Cisco Nexus 3600 System Management Configuration Guide |