



Configure object tracking

This module describes the configuration of object tracking on your Cisco IOS XR network. To locate documentation for other commands that might appear in the course of performing a configuration task, search online in *Cisco ASR 9000 Series Aggregation Services Router Commands Master List*.

Table 1: Feature History Table

Feature Name	Release Information	Feature Description
Enhanced object tracking for loopback interfaces	Release 7.10.1	You can now track an object (for example, an interface) and configure to error-disable a loopback interface when the tracked object state changes. This is beneficial when the loopback interface is the source of multiple tunnels and all tunnels need to be brought down simultaneously. Earlier, you couldn't error-disable loopback interface.
Enhanced object tracking	Release 6.4.2	The Enhanced Object Tracking feature is introduced. The ability to error-disable interfaces is added based on the state of objects that are tracked.
Enhanced object tracking	Release 4.2.1	The ability to create a tracked list based on a threshold percentage or weight was added.
Enhanced object tracking	Release 4.0.0	This feature was introduced.

- [Information about object tracking, on page 2](#)
- [Implement object tracking, on page 3](#)
- [Enhanced object tracking, on page 5](#)

Information about object tracking

Object tracking enables the system to monitor a specific object and perform a client-defined action on another object. The object receiving the action may not have any direct relationship with the tracked object. The system bases these actions on changes in the properties of the tracked object.

- **Boolean AND function** — When you assign a Boolean AND function to a tracked list, the system requires every object in the subset to be in the "up" state for the tracked object to be considered "up."
- **Boolean OR function** — When you assign a Boolean OR function to a tracked list, the system requires at least one object in the subset to be in the "up" state for the tracked object to be considered "up."

Assign a tracked object

You can assign each tracked object a **unique name** using the `track` command in **configuration mode**.

Track process

The tracking process operates as follows:

- **Polling**: The system periodically polls the tracked object.
- **State Monitoring**: It detects and reports any changes in the object's state.
- **State Values**: The tracked object can be in either an "up" or "down" state.
- **Polling Timing**: Polling occurs either:
 - **Immediately**, or
 - **After a configured delay**

Track multiple objects

You can track multiple objects using a **list**. The system supports flexible combinations using **Boolean logic**.

Boolean AND Function

- When you assign a **Boolean AND** function to a tracked list:
 - **All** objects in the subset must be in the "up" state.
 - This condition sets the tracked object to the "up" state.

Boolean OR Function

- When you assign a **Boolean OR** function to a tracked list:
 - **At least one** object in the subset must be in the "up" state.
 - This condition sets the tracked object to the "up" state.

Implement object tracking

The purpose of this task is to implement object tracking and guide you on various object tracking procedures.

Before you begin

- You must be in a user group associated with a task group that includes the proper task IDs.
- The command reference guides include the task IDs required for each command.
- If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
- Steps listed below can be performed in any order based on your requirement.

Procedure

Step 1 **Track the line protocol state of an interface** - To track the line protocol state of an interface in global configuration mode on a Cisco device, you can use the object tracking feature:

Example:

```
Router# configure
Router(config)# track track1
Router(config-track)# type line-protocol state
Router(config-track-line-prot)# interface atm 0/2/0/0.1
Router(config-track-line-prot)# exit
Router(config-track)# delay up 10 down 5 (optional)
Router(config-track)# commit
```

Step 2 **Track the IP route reachability** - To track the state of a route in the routing table using Cisco's object tracking, you can configure a tracked object that monitors whether a specific route exists and is reachable. You can do it in the global configuration mode:

Example:

```
Router# configure
Router(config)# track track1
Router(config-track)# type route reachability vrf internet
Router(config-track-route)# vrf vrf-table-4
Router(config-track-line-prot)# exit
Router(config-track)# delay up 10
Router(config-track)# commit
```

You can configure the type of IP route to be tracked, which can consist of either of the following, depending on your router type:

- Router(config-track-route)# vrf vrf-table-4 or
- Router(config-track-route)# route ipv4 10.56.8.10/16

Step 3 **Build a track based on a list of objects** - To create a tracked list of objects using the boolean expression in global configuration mode on a Cisco device, follow these steps. This is useful when you want to track the combined state of multiple interfaces or routes using logical operators like AND, OR, and NOT.

Note

An object must exist before it can be added to a tracked list. The `NOT` operator is specified for one or more objects and it negates the state of the object.

Example:

```
Router# configure
Router(config)# track track1
Router(config-track-list)# type list boolean and
Router(config-track-list)# object 3 not
Router(config-track-line-prot)# exit
Router(config-track)# delay up 10
Router(config-track)# commit
```

- Step 4** **Build a track based on a list of objects - threshold percentage** - To create a tracked list of objects using a threshold percentage in global configuration mode, use the `threshold percentage` option. This allows the tracked list to be considered **up** when a specified percentage of the tracked objects are up.

Example:

```
Router# configure
Router(config)# track track1
Router(config-track-list)# type list threshold percentage
Router(config-track-list-threshold)# object 1
Router(config-track-list-threshold)# object 2
Router(config-track-list-threshold)# object 3
Router(config-track-list-threshold)# object 4
Router(config-track-list-threshold)# threshold percentage up 50 down 33
Router(config-track)# commit
```

- Step 5** **Build a track based on a list of objects - threshold weight** - To create a tracked list of objects using a threshold weight in global configuration mode, follow this configuration. This method allows you to assign weights to individual tracked objects and determine the overall state of the list based on a cumulative weight threshold.

Example:

```
Router# configure
Router(config)# track track1
Router(config-track-list)# type list threshold weight
Router(config-track-list-threshold)# object 1 weight 10
Router(config-track-list-threshold)# object 2 weight 5
Router(config-track-list-threshold)# object 3 weight 3
Router(config-track-list-threshold)# threshold weight up 10 down 5
Router(config-track)# commit
```

- Step 6** **Track IP SLA reachability** - To enable the tracking of the return code of an IP SLA operation in global configuration mode on a Cisco device, you can configure an object to monitor the success or failure of an IP SLA probe. This is useful for tracking network performance or availability.

Example:

```
Router# configure
Router(config)# track track1
Router(config-track)# type rtr 100 reachability
Router(config-track)# delay up 5
Router(config-track)# delay down 10
Router(config-track)# commit
```

Enhanced object tracking

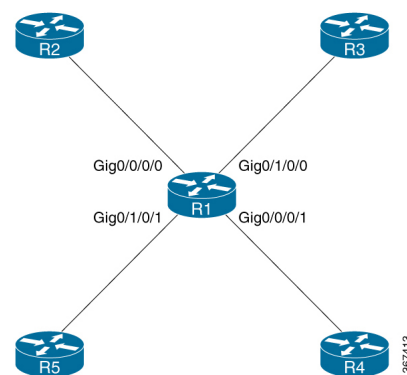
Enhanced Object Tracking lets you extend tracking functions and implement actions. Actions trigger when the tracked object moves from **up** to **down** states.

Based on the track state, you can error-disable one or more specified interfaces. If you don't configure the **auto-recover** keyword, the interfaces stay disabled even after the track state returns to its original state. You can configure **auto-recover** for each **action** configuration associated with the track.

In the Figure, router R1 configures tracks named track1 and track2 to track the line protocol state of interfaces GigabitEthernet0/0/0/1 and GigabitEthernet0/1/0/1, respectively. Track3 tracks both track1 and track2 using Boolean logic AND, so if either track1 or track2 goes down, track3 also goes down.

Additionally, track3 uses the action command to disable interfaces GigabitEthernet0/0/0/0 and GigabitEthernet0/1/0/0 when track3 goes down.

Figure 1: Enhanced Object Tracking



Once the interfaces are error-disabled, they remain in the error-disabled state even if the track state changes to the **up** state. This is the default behavior. To change this default behavior, you can optionally configure the **auto-recover** keyword in the **action** command.

If you configure the optional **auto-recover** keyword, then the error-disabled state on the interfaces is cleared when the track state changes to the **up** state.

Configure enhanced object tracking

The purpose of this task is to configure the Enhanced Object Tracking using the `action` command. You must first

- define a **track object** and
- then specify actions to be taken when the state of that object changes, for example from **up** to **down** or vice versa.

Before you begin

- Ensure the Enhanced Object Tracking is supported and enabled on your platform.
- The tracked object, such as interface, IP SLA, and route must be configured before applying the `action` commands.

Procedure

Step 1 Configure track1 to track line-protocol state of the interface FourHundredGigE0/0/0/1

Example:

```
Router#configure
Router(config)#track track1
Router(config-track)#type line-protocol state
Router(config-track-line-prot)#interface FourHundredGigE0/0/0/1
Router(config-track-line-prot)#exit
Router(config-track)#exit
```

Step 2 Configure track2 to track line-protocol state of the interface FourHundredGigE0/1/0/1

Example:

```
Router(config)#track track2
Router(config-track)#type line-protocol state
Router(config-track-line-prot)#interface FourHundredGigE0/1/0/1
Router(config-track-line-prot)#exit
Router(config-track)#exit
```

Step 3 Configure track3 with boolean AND of track1 state and track2 state. Specify actions to take when track3 state changes.

Example:

```
Router(config)#track track3
Router(config-track)#type list boolean and
Router(config-track-list-boolean)#object track1
Router(config-track-list-boolean)#object track2
Router(config-track-line-boolean)#exit
Router(config-track)#action
Router(config-track-action)#track-down error-disable interface FourHundredGigE0/0/0/0 auto-recover
Router(config-track-action)#track-down error-disable interface FourHundredGigE0/1/0/0
```

Step 4 Configure track4 to track line-protocol state of the HundredGigE0/0/0/35 interface and take action on loopback interface.

Example:

```
Router(config)#track track4
Router(config-track)#type line-protocol state
Router(config-track-line-prot)#interface HundredGigE0/0/0/35
Router(config-track-line-prot)#exit
```

Step 5 Specify action to take when track4 state changes. In this example, the action happens when the track state changes to up.

Example:

```
Router(config-track)#action
Router(config-track-action)#track-up error-disable interface Loopback100 auto-recover
Router(config-track)#exit
Router(config)#end
```

Verify enhanced object tracking

The purpose of this task is to use the **show track** command and view the state of the track.

Procedure

Step 1 Use this state information to configure a **tracked list** using Boolean logic, threshold percentage, or threshold weight, depending on your desired behavior.

Example:

```
Router#show track
Track track3
    List boolean and is UP
    7 changes, last change 16:04:28 IST Mon Jul 02 2018
    object track2 UP
    object track1 UP
Track track1
    Interface FourHundredGigE0/0/0/1 line-protocol
    Line protocol is UP
    7 changes, last change 16:04:28 IST Mon Jul 02 2018
Track track2
    Interface FourHundredGigE0/1/0/1 line-protocol
    Line protocol is UP
    7 changes, last change 16:02:41 IST Mon Jul 02 2018

Track track4
    Interface HundredGigE0/0/0/35 line-protocol
    Line protocol is DOWN
    2 changes, last change 06:28:06 UTC Tue Jun 27 2023
    Delay up 0 secs(default), down 0 secs(default)
```

Step 2 Use the **show errdisable** command to verify whether the interface configured for tracking is error-disabled. Since none of the track states match the track-action state, the system places no interfaces in an error-disabled state.

Example:

```
Router#show error-disable
Interface          Error-Disable reason          Retry (s)  Time disabled
-----
There are no interfaces error-disabled matching the given criteria
```

To view the status of all the interfaces of the tracked object, use the **show interface brief** command.

```
Router#show interface brief
Intf Name          Intf State LineP State  Encap Type  MTU (byte) BW (Kbps)
-----
Lo100              up         up          Loopback    1500        0
FourHundredGigE0/0/0/0 up         up          ARPA        1514        100000000
FourHundredGigE0/0/0/1 up         up          ARPA        1514        100000000
FourHundredGigE0/1/0/0 up         up          ARPA        1514        100000000
FourHundredGigE0/1/0/1 up         up          ARPA        1514        100000000
HundredGigE0/0/0/35 admin-down admin-down  ARPA        1514        100000000
```

Step 3 When a track state changes, the system executes the corresponding track action and updates the status of the interfaces configured in that action.

- Track 3 enters the **down** state when either Track 1 or Track 2 goes down.
- Track 4 enters the **up** state when the HundredGigE0/0/0/35 interface comes up.
- When Track 3 is **down** and Track 4 is **up**, the **show error-disable** command displays the resulting output.

Example:

```
Router#show error-disable
Interface          Error-Disable reason          Retry (s)  Time disabled
-----
Loopback100        ot-track-state-change         ---        08:44:07 FH0/0/0/0
ot-track-state-change ---        08:42:08 FH0/1/0/0        ot-track-state-change
---                08:42:01
```

- When track3 state is **down** and track4 state is **up**, the **show interface brief** command displays the following output.

Example:

```
Router#show interface brief
Intf Name          Intf State  LineP State  Encap Type  MTU (byte)  BW (Kbps)
-----
Lo100              err-disable admin-down   Loopback    1500        0
FourHundredGigE0/0/0/0 err-disable admin-down   ARPA        1514        100000000
FourHundredGigE0/0/0/1 err-disable admin-down   ARPA        1514        100000000
FourHundredGigE0/1/0/0 err-disable admin-down   ARPA        1514        100000000
FourHundredGigE0/1/0/1 up           up           ARPA        1514        100000000
HundredGigE0/0/0/35 up           up           ARPA        1514        100000000
```

Step 4 Run the **show interface brief** command to view interface details.

when track3 state comes back up, the error-disable status on the interface FourHundredGigE0/0/0/0 clears. This is because of the **auto-recover** configuration for FourHundredGigE0/0/0/0. However, interface FourHundredGigE0/1/0/0 remains in the error-disable status because **auto-recover** is not configured on this interface.

Similarly, when track4 state goes down, the error-disable status on the interface Loopback100 clears because of the **auto-recover** configuration for track4.

Example:

```
RP/0/0/CPU0:ios#show interface brief
Intf Name          Intf State  LineP State  Encap Type  MTU (byte)  BW (Kbps)
-----
Lo100              up          up           Loopback    1500        0
FourHundredGigE0/0/0/0 up          up           ARPA        1514        100000000
FourHundredGigE0/0/0/1 up          up           ARPA        1514        100000000
FourHundredGigE0/1/0/0 err-disable admin-down   ARPA        1514        100000000
FourHundredGigE0/1/0/1 up          up           ARPA        1514        100000000
HundredGigE0/0/0/35 admin-down  admin-down   ARPA        1514        100000000
```