



Implementing Object Tracking on Cisco IOS XR Software

This module describes the configuration of object tracking on your Cisco IOS XR network. For information about its application in IPSec, see *Cisco IOS XR System Security Configuration Guide for the Cisco XR 12000 Series Router*.

For complete descriptions of the commands listed in this module, see [Related Documents, page 13](#). To locate documentation for other commands that might appear in the course of performing a configuration task, search online in *Cisco IOS XR Commands Master List for the Cisco XR 12000 Series Router*.

Table 1: Feature History for Implementing Object Tracking on Cisco IOS XR Software Contents

Release	Modification
Release 3.6.0	This feature was introduced.
Release 3.7.0	No modification.
Release 3.8.0	No modification.
Release 3.9.0	No modification.

This module contains the following topics:

- [Prerequisites for Implementing Object Tracking, page 2](#)
- [Information About Object Tracking, page 2](#)
- [How to Implement Object Tracking, page 2](#)
- [Configuration Examples for Configuring Object Tracking, page 11](#)
- [Additional References, page 13](#)

Prerequisites for Implementing Object Tracking

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.

Information About Object Tracking

Object tracking is a mechanism to track an object and to take an action on another object with no relationship to the tracked objects, based on changes to the properties of the object being tracked.

Each tracked object is identified by a unique name specified on the tracking command-line interface (CLI). Cisco IOS XR processes then use this name to track a specific object.

The tracking process periodically polls the tracked object and reports any changes to its state in terms of its being up or down, either immediately or after a delay, as configured by the user.

Multiple objects can also be tracked by means of a list, using a flexible method for combining objects with Boolean logic. This functionality includes:

- **Boolean AND function**—When a tracked list has been assigned a Boolean AND function, each object defined within a subset must be in an up state, so that the tracked object can also be in the up state.
- **Boolean OR function**—When the tracked list has been assigned a Boolean OR function, it means that at least one object defined within a subset must also be in an up state, so that the tracked object can also be in the up state.

How to Implement Object Tracking

This section describes the various object tracking procedures.

Tracking Whether an Interface Is Up or Down

Perform this optional task in global configuration mode to track, in increments of from 1 to 10 seconds, whether the state of an interface is up or down.

When the tracked object state changes to down, the tracking object (in the case of IPSec, this is the service virtual interface [SVI]) is brought down, which results in the following:

- All existing tunnels on the SVI are torn down.
- New tunnels cannot be established on this SVI.
- All the routes, whether static or dynamic, pointing to the SVI are removed, including reverse-route injections (RRI).

SUMMARY STEPS

1. **configure**
2. **track** *track-name*
3. (Optional) **delay** { *up seconds* | *down seconds* }
4. Use one of the following commands:
 - **end**
 - **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/0/CPU0:router# configure	Enters global configuration mode.
Step 2	track <i>track-name</i> Example: RP/0/0/CPU0:router(config)# track track1	Enters track configuration mode. <ul style="list-style-type: none"> • <i>track-name</i> —Specifies a name for the object to be tracked.
Step 3	delay { <i>up seconds</i> <i>down seconds</i> } Example: RP/0/0/CPU0:router(config-track)# delay up 10	(Optional) Schedules the delay that can occur between tracking whether the object is up or down.
Step 4	Use one of the following commands: <ul style="list-style-type: none"> • end • commit Example: RP/0/0/CPU0:router(config-track)# end or RP/0/0/CPU0:router(config-track)# commit	Saves configuration changes. <ul style="list-style-type: none"> • When you issue the end command, the system prompts you to commit changes: Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]: <ul style="list-style-type: none"> ◦ Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. ◦ Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes. ◦ Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.

	Command or Action	Purpose
		<ul style="list-style-type: none"> Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.

Tracking the Line Protocol State of an Interface

Perform this task in global configuration mode to track the line protocol state of an interface.

A tracked object is considered up when a line protocol of the interface is up.

After configuring the tracked object, you may associate the interface whose state should be tracked and specify the number of seconds to wait before the tracking object polls the interface for its state.

SUMMARY STEPS

- configure**
- track** *track-name*
- type line-protocol state**
- interface** *type interface-path-id*
- (Optional) **delay** { **up** *seconds* | **down** *seconds* }
- interface** { **service-gre** *numeric-name* | **service-ipsec** *numeric-name* }
- line-protocol track** *object-name*
- Use one of the following commands:
 - end**
 - commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/0/CPU0:router# configure	Enters global configuration mode.
Step 2	track <i>track-name</i> Example: RP/0/0/CPU0:router(config)# track track1	Enters track configuration mode. <ul style="list-style-type: none"> <i>track-name</i> —Specifies a name for the object to be tracked.

	Command or Action	Purpose
Step 3	type line-protocol state Example: RP/0/0/CPU0:router(config-track)# type line-protocol state	Creates a track based on the line protocol of an interface.
Step 4	interface type interface-path-id Example: RP/0/0/CPU0:router(config-track-line-prot)#interface atm 0/2/0/0.1	Enters interface configuration mode. <ul style="list-style-type: none"> • <i>type</i> —Specifies the interface type. For more information, use the question mark (?) online help function. • <i>interface-path-id</i> —Identifies a physical interface or a virtual interface. <p>Note Use the show interfaces command to see a list of all possible interfaces currently configured on the router.</p> <p>Note The loopback and null interfaces are always in the up state and, therefore, cannot be tracked.</p>
Step 5	delay { up seconds down seconds } Example: RP/0/0/CPU0:router(config-track)# delay up 10	(Optional) Schedules the delay that can occur between tracking whether the object is up or down.
Step 6	interface { service-gre numeric-name service-ipsec numeric-name } Example: RP/0/0/CPU0:router(config-track)# interface service-ipsec 23	Enters the service-ipsec interface mode, in which you associate a service-ipsec interface with the interface whose state should be tracked. For example, if the state of the selected interface, such as, ATM, goes down or up, the state of the service-ipsec interface follows suit. <ul style="list-style-type: none"> • <i>numeric-name</i> —Numeric name of the service-ipsec interface, which can be from 1-65535. <p>Note Although service-gre interfaces can be tracked as an interface object, it is currently unsupported as a means to monitor the state of another interface object.</p>
Step 7	line-protocol track object-name Example: RP/0/0/CPU0:router(config-if)# line-protocol track track12	Associates a specific track to an IP Sec or GRE interface. The state of the interface changes when the state of the track changes.
Step 8	Use one of the following commands:	Saves configuration changes.

Command or Action	Purpose
<ul style="list-style-type: none"> • end • commit <p>Example:</p> <pre>RP/0/0/CPU0:router(config-track)# end OR RP/0/0/CPU0:router(config-track)# commit</pre>	<ul style="list-style-type: none"> • When you issue the end command, the system prompts you to commit changes: <ul style="list-style-type: none"> Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]: ◦ Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. ◦ Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes. ◦ Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.

Tracking IP Route Reachability

When a host or a network goes down on a remote site, routing protocols notify the router and the routing table is updated accordingly. The routing process is configured to notify the tracking process when the route state changes due to a routing update.

A tracked object is considered up when a routing table entry exists for the route and the route is accessible.

SUMMARY STEPS

1. **configure**
2. **track** *track-name*
3. **type route reachability**
4. Use one of the following commands:
 - **vrf** *vrf-table-name*
 - **route ipv4** *IP-prefix/mask*
5. (Optional) **delay** { **up** *seconds* | **down** *seconds* }
6. **interface** { **service-gre** *numeric-name* | **service-ipsec** *numeric-name* }
7. **line-protocol track** *object-name*
8. Use one of the following commands:
 - **end**
 - **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/0/CPU0:router# configure	Enters global configuration mode.
Step 2	track <i>track-name</i> Example: RP/0/0/CPU0:router(config)# track track1	Enters track configuration mode. • <i>track-name</i> —Specifies a name for the object to be tracked.
Step 3	type route reachability Example: RP/0/0/CPU0:router(config-track)# type route reachability vrf internet	Configures the routing process to notify the tracking process when the state of the route changes due to a routing update.
Step 4	Use one of the following commands: <ul style="list-style-type: none"> • vrf <i>vrf-table-name</i> • route ipv4 <i>IP-prefix/mask</i> 	Configures the type of IP route to be tracked, which can consist of either of the following, depending on your router type: <ul style="list-style-type: none"> • <i>vrf-table-name</i> —A VRF table name. • <i>IP-prefix/mask</i> —An IP prefix consisting of the network and subnet mask (for example, 10.56.8.10/16).

	Command or Action	Purpose
	<p>Example:</p> <pre>RP/0/0/CPU0:router(config-track-route)# vrf vrf-table-4 or RP/0/0/CPU0:router(config-track-route)# route ipv4 10.56.8.10/16</pre>	
Step 5	<p>delay { up <i>seconds</i> down <i>seconds</i> }</p> <p>Example:</p> <pre>RP/0/0/CPU0:router(config-track)# delay up 10</pre>	<p>(Optional) Schedules the delay that can occur between tracking whether the object is up or down.</p>
Step 6	<p>interface { service-gre <i>numeric-name</i> service-ipsec <i>numeric-name</i> }</p> <p>Example:</p> <pre>RP/0/0/CPU0:router(config-track)# interface service-ipsec 23</pre>	<p>Enters the service-ipsec interface mode, in which you associate a service-ipsec interface with the interface whose state should be tracked. For example, if the state of the selected interface, such as, ATM, goes down or up, the state of the service-ipsec interface follows suit.</p> <ul style="list-style-type: none"> • <i>numeric-name</i> —Numeric name of the service-ipsec interface, which can be from 1-65535. <p>Note Although service-gre interfaces can be tracked as an interface object, it is currently unsupported as a means to monitor the state of another interface object.</p>
Step 7	<p>line-protocol track <i>object-name</i></p> <p>Example:</p> <pre>RP/0/0/CPU0:router(config-if)# line-protocol track track1</pre>	<p>Associates the track with an IPSec or GRE interface. The state of the interface changes when the state of the track changes.</p>
Step 8	<p>Use one of the following commands:</p> <ul style="list-style-type: none"> • end • commit <p>Example:</p> <pre>RP/0/0/CPU0:router(config-if)# end or RP/0/0/CPU0:router(config-if)# commit</pre>	<p>Saves configuration changes.</p> <ul style="list-style-type: none"> • When you issue the end command, the system prompts you to commit changes: <pre>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</pre> <ul style="list-style-type: none"> ◦ Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. ◦ Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes. ◦ Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.

	Command or Action	Purpose
		<ul style="list-style-type: none"> • Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.

Building a Track Based on a List of Objects

Perform this task in the global configuration mode to create a tracked list of objects (which, in this case, are lists of interfaces or prefixes) using a Boolean expression to determine the state of the list.

A tracked list contains one or more objects. The Boolean expression enables two types of calculations 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.



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 negates the state of the object.

After configuring the tracked object, you must associate the interface whose state should be tracked and you may optionally specify the number of seconds to wait before the tracking object polls the interface for its state.

SUMMARY STEPS

1. **configure**
2. **track** *track-name*
3. **type list boolean** { **and** | **or** }
4. **object** *object-name* [**not**]
5. (Optional) **delay** { **up** *seconds* | **down** *seconds* }
6. **interface** { **service-gre** *numeric-name* | **service-ipsec** *numeric-name* }
7. **line-protocol track** *object-name*
8. Use one of the following commands:
 - **end**
 - **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure Example: RP/0/0/CPU0:router# configure	Enters global configuration mode.
Step 2	track track-name Example: RP/0/0/CPU0:router(config)# track track1	Enters track configuration mode. <ul style="list-style-type: none"> • <i>track-name</i> —Specifies a name for the object to be tracked.
Step 3	type list boolean { and or } Example: RP/0/0/CPU0:router(config-track-list)# type list boolean and	Configures a Boolean list object and enters track list configuration mode. <ul style="list-style-type: none"> • boolean —Specifies that the state of the tracked list is based on a Boolean calculation. • 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. • 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.
Step 4	object object-name [not] Example: RP/0/0/CPU0:router(config-track-list)# object 3 not	Specifies the object to be tracked by the list <ul style="list-style-type: none"> • <i>object-name</i> —Name of the object to track. • not —Negates the state of the object.
Step 5	delay { up seconds down seconds } Example: RP/0/0/CPU0:router(config-track)# delay up 10	(Optional) Schedules the delay that can occur between tracking whether the object is up or down.
Step 6	interface { service-gre numeric-name service-ipsec numeric-name } Example: RP/0/0/CPU0:router(config-track)# interface service-ipsec 23	Enters the service-ipsec interface mode, in which you associate a service-ipsec interface with the interface whose state should be tracked. For example, if the state of the selected interface, such as, ATM, goes down or up, the state of the service-ipsec interface follows suit. <ul style="list-style-type: none"> • <i>numeric-name</i> —Numeric name of the service-ipsec interface, which can be from 1-65535. <p>Note Although service-gre interfaces can be tracked as an interface object, it is currently unsupported as a means to monitor the state of another interface object.</p>

	Command or Action	Purpose
Step 7	<p>line-protocol track <i>object-name</i></p> <p>Example:</p> <pre>RP/0/0/CPU0:router(config-if)# line-protocol track track1</pre>	<p>Associates the track to an IP Sec or GRE interface. The state of the interface changes when the state of the track changes.</p>
Step 8	<p>Use one of the following commands:</p> <ul style="list-style-type: none"> • end • commit <p>Example:</p> <pre>RP/0/0/CPU0:router(config-track)# end OR RP/0/0/CPU0:router(config-track)# commit</pre>	<p>Saves configuration changes.</p> <ul style="list-style-type: none"> • When you issue the end command, the system prompts you to commit changes: <pre>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</pre> <ul style="list-style-type: none"> ◦ Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. ◦ Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes. ◦ Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuration Examples for Configuring Object Tracking

For examples illustrating how to use object tracking in a variety of scenarios in IPSec, see the *Implementing IPSec Network Security on Cisco IOS XR Software* module in *Cisco IOS XR System Monitoring Configuration Guide for the Cisco XR 12000 Series Router*.

Tracking Whether the Interface Is Up or Down: Example

```
track connection100
  type list boolean and
  object object3 not
  delay up 10
!
interface service-ipsec 23
  line-protocol track connection100
!
```

Tracking the Line Protocol State of an Interface: Example

In this example, traffic arrives from interface service-ipsec1 and exits through interface GigabitEthernet 0/0/0/3:

```
track IPsec1
  type line-protocol state
  interface gigabitethernet0/0/0/3
  !
interface service-ipsec 1
  ipv4 address 70.0.0.1 255.255.255.0
  profile vrf1 profile ipsec
  line-protocol track IPsec1
  tunnel source 80.0.0.1
  tunnel destination 80.0.0.2
  service-location preferred-active 0/0/1
  !
```

Displaying the Line Protocol State of an Interface: Example

This example displays the output from the **show track** command after performing the previous example:

```
RP/0/0/CPU0:router# show track

Track IPsec1
Interface GigabitEthernet0_0_0_3 line-protocol
!
  Line protocol is UP
  1 change, last change 10:37:32 UTC Thu Sep 20 2007
Tracked by:
  service-ipsec1
  !
```

Tracking IP Route Reachability: Example

In this example, traffic arriving from interface service-ipsec1 has its destination in network 7.0.0.0/24. This tracking procedure follows the state of the routing protocol prefix to signal when there are changes in the routing table.

```
track PREFIX1
  type route reachability
  route ipv4 7.0.0.0/24
  !
interface service-ipsec 1
  vrf 1
  ipv4 address 70.0.0.2 255.255.255.0
  profile vrf_1_ipsec
  line-protocol track PREFIX1
  tunnel source 80.0.0.2
  tunnel destination 80.0.0.1
  service-location preferred-active 0/2/0
```

Building a Track Based on a List of Objects: Example

In this example, traffic arriving from interface service-ipsec1 exits through interface GigabitEthernet 0/0/0/3 and interface ATM 0/2/0/0.1. The destination of the traffic is at network 7.0.0.0/24.

If either one of the interfaces or the remote network goes down, the flow of traffic must stop. To do this, we use a Boolean AND expression.

```
track LIST2
  type list boolean and
```

```

    object IPsec1
    object IPsec2
    object PREFIX1
    !
track IPsec1
  type line-protocol state
  interface GigabitEthernet0/0/0/3
  !
track IPsec2
  type line-protocol state
  interface ATM0/2/0.1
  !
track PREFIX1
  type route reachability
  route ipv4 7.0.0.0/24
  !
interface service-ipsec1
  vrf 1
  ipv4 address 70.0.0.2 255.255.255.0
  profile vrf_1_ipsec
  line-protocol track LIST2
  tunnel source 80.0.0.2
  tunnel destination 80.0.0.1
  service-location preferred-active 0/2/0
  !

```

Additional References

The following sections provide references related to implementing object tracking for IPsec network security.

Related Documents

Related Topic	Document Title
IPsec network security commands	<i>IPsec Network Security Commands on the Cisco IOS XR Software module in Cisco IOS XR System Security Configuration Guide for the Cisco XR 12000 Series Router</i>
Internet Key Exchange (IKE) security protocol commands	<i>Internet Key Exchange Security Protocol Commands on the Cisco IOS XR Software module in Cisco IOS XR System Security Command Reference for the Cisco XR 12000 Series Router</i>
IP-Sec-related object tracking commands	<i>Cisco IOS XR System Management Command Reference for the Cisco XR 12000 Series Router</i>

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
—	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
RFC 2401	<i>Security Architecture for the Internet Protocol</i>

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport