# Configuring Enhanced Object Tracking

**First Published: May 2, 2005**
**Last Updated: November 24, 2010**

Before the introduction of the Enhanced Object Tracking feature, the Hot Standby Router Protocol (HSRP) had a simple tracking mechanism that allowed you to track the interface line-protocol state only. If the line-protocol state of the interface went down, the HSRP priority of the router was reduced, allowing another HSRP router with a higher priority to become active.

The Enhanced Object Tracking feature separates the tracking mechanism from HSRP and creates a separate standalone tracking process that can be used by other Cisco IOS processes as well as HSRP. This feature allows tracking of other objects in addition to the interface line-protocol state.

A client process, such as HSRP, Virtual Router Redundancy Protocol (VRRP), or Gateway Load Balancing Protocol (GLBP), can now register its interest in tracking objects and then be notified when the tracked object changes state.

## Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "Feature Information for Enhanced Object Tracking" section on page 31.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

## Contents



**Americas Headquarters:**
**Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA**

# Restrictions for Enhanced Object Tracking

Enhanced Object Tracking is not stateful switchover (SSO)-aware and cannot be used with Hot Standby Routing Protocol (HSRP), Virtual Router Redundancy Protocol (VRRP), or Gateway Load Balancing Protocol (GLBP) in SSO mode.

# Information About Enhanced Object Tracking



## Feature Design of Enhanced Object Tracking

Enhanced Object Tracking provides complete separation between the objects to be tracked and the action to be taken by a client when a tracked object changes. Thus, several clients such as HSRP, VRRP, or GLPB can register their interest with the tracking process, track the same object, and each take different action when the object changes.

Each tracked object is identified by a unique number that is specified on the tracking command-line interface (CLI). Client processes use this number to track a specific object.

The tracking process periodically polls the tracked objects and notes any change of value. The changes in the tracked object are communicated to interested client processes, either immediately or after a specified delay. The object values are reported as either up or down.

You can also configure a combination of tracked objects in a list and a flexible method for combining objects using Boolean logic. This functionality includes the following capabilities:

- Threshold—The tracked list can be configured to use a weight or percentage threshold to measure the state of the list. Each object in a tracked list can be assigned a threshold weight. The state of the tracked list is determined by whether or not the threshold has been met.
- Boolean “and” function—When a tracked list has been assigned a Boolean “and” function, it means that each object defined within a subset must be in an up state so that the tracked object can become up.
- 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 be in an up state so that the tracked object can become up.

As of Cisco IOS Release 15.1(3)T, a maximum of 1000 objects can be tracked. Although 1000 tracked objects can be configured, each tracked object uses CPU resources. The amount of available CPU resources on a router is dependent upon variables such as traffic load and how other protocols are configured and run. The ability to use 1000 tracked objects is dependent upon the available CPU. Testing should be conducted on site to ensure that the service works under the specific site traffic conditions.

# Scaled Route Metrics

The **track ip route** command enables tracking of a route in the routing table. If a route exists in the table, the metric value is converted into a number. To provide a common interface to tracking clients, route metric values are normalized to the range from 0 to 255, where 0 is connected and 255 is inaccessible. Scaled metrics can be tracked by setting thresholds. Up and down state notification occurs when the thresholds are crossed. The resulting value is compared against threshold values to determine the tracking state as follows:

- State is up if the scaled metric for that route is less than or equal to the up threshold.
- State is down if the scaled metric for that route is greater than or equal to the down threshold.

Tracking uses a per-protocol configurable resolution value to convert the real metric to the scaled metric. Table 1 shows the default values used for the conversion. You can use the **track resolution** command to change the metric resolution default values.

***Table 1 Metric Conversion***

| Route Type[1] | Metric Resolution |
|---|---|
| Static | 10 |
| Enhanced Interior Gateway Routing Protocol (EIGRP) | 2560 |
| Open Shortest Path First (OSPF) | 1 |
| Intermediate System-to-Intermediate System (IS-IS) | 10 |

1. RIP is scaled directly to the range from 0 to 255 because its maximum metric is less than 255.

For example, a change in 10 in an IS-IS metric results in a change of 1 in the scaled metric. The default resolutions are designed so that approximately one 2-Mbps link in the path will give a scaled metric of 255.

Scaling the very large metric ranges of EIGRP and IS-IS to a 0 to 255 range is a compromise. The default resolutions will cause the scaled metric to go above the maximum limit with a 2-Mbps link. However, this scaling allows a distinction between a route consisting of three Fast-Ethernet links and a route consisting of four Fast-Ethernet links.

# IP SLA Operation Tracking

Object tracking of IP SLAs operations allows tracking clients to track the output from IP SLAs objects and use the provided information to trigger an action.

Cisco IOS IP SLAs is a network performance measurement and diagnostics tool that uses active monitoring. Active monitoring is the generation of traffic in a reliable and predictable manner to measure network performance. Cisco IOS software uses IP SLAs to collect real-time metrics such as response time, network resource availability, application performance, jitter (interpacket delay variance), connect time, throughput, and packet loss.

These metrics can be used for troubleshooting, for proactive analysis before problems occur, and for designing network topologies.

Every IP SLAs operation maintains an operation return-code value. This return code is interpreted by the tracking process. The return code can return OK, OverThreshold, and several other return codes. Different operations can have different return-code values, so only values common to all operation types are used.

Two aspects of an IP SLAs operation can be tracked: state and reachability. The difference between these aspects relates to the acceptance of the OverThreshold return code. Table 2 shows the state and reachability aspects of IP SLAs operations that can be tracked.

***Table 2 Comparison of State and Reachability Operations***

| Tracking | Return Code | Track State |
|---|---|---|
| State | OK | Up |
| | (all other return codes) | Down |
| Reachability | OK or OverThreshold | Up |
| | (all other return codes) | Down |

# Enhanced Object Tracking and Embedded Event Manager

Beginning with Cisco IOS Release 12.4(2)T, Enhanced Object Tracking (EOT) is now integrated with Embedded Event Manager (EEM) to allow EEM to report on status change of a tracked object and to allow EOT to track EEM objects. A new type of tracking object—a stub object—is created. The stub object can be modified by an external process through a defined Application Programming Interface (API). See the "Embedded Event Manager Overview" document in the *Cisco IOS Network Management Configuration Guide* for more information on how EOT works with EEM.

# EOT Support for Carrier Delay

The EOT Support for Carrier Delay feature enables Enhanced Object Tracking (EOT) to consider the carrier-delay timer when tracking the status of an interface.

If a link fails, by default there is a two-second timer that must expire before an interface and the associated routes are declared as being down. If a link goes down and comes back up before the carrier delay timer expires, the down state is effectively filtered, and the rest of the software on the switch is not aware that a link-down event occurred. You can configure the carrier-delay seconds command in interface configuration mode to extend the timer up to 60 seconds.

When EOT is configured on an interface, the tracking may detect the interface is down before a configured carrier-delay timer has expired. This is because EOT looks at the interface state and does not consider the carrier delay timer. Use the **carrier-delay** command in tracking configuration mode to enable tracking to consider the carrier-delay timer configured on an interface.

## Enhanced Object Tracking for Mobile IP Applications

The Enhanced Object Tracking Support for Mobile IP feature enables EOT to monitor the presence of Home Agent, Packet Data Serving Node (PDSN), or Gateway GPRS Support Node (GGSN) traffic on a router for mobile wireless applications.

When a redundant pair of Home Agents running HSRP between them loses connectivity, both HSRP nodes become active. Once the connectivity is restored between the two nodes, a graceful way is needed to restore proper HSRP states without losing Home Agent bindings. During the time of no connectivity, one of the nodes will continue to process Home Agent, GGSN, or PDSN traffic while the other will not. The node that continues to process traffic needs to remain active once connectivity is restored. To ensure that the active node remains in the active state, the priority of the HSRP group member that does not process Home Agent traffic is reduced. Reducing the priority of the node that is not processing Home Agent traffic ensures that this node will become the standby after connectivity is restored. When connectivity is restored, the normal Home Agent state synchronization will get all bindings back into the inactive node and, depending on the preempt configuration, it may switch over again. This state synchronization ensures that no Mobile IP, GGSN, or PDSN bindings are lost.

For more information on configuring Mobile IP services, see the following Cisco IOS configuration guides:

- *Cisco IOS Mobile Wireless Home Agent Configuration Guide*
- *Cisco IOS Mobile Wireless Gateway GPRS Support Node Configuration Guide*
- *Cisco IOS Mobile Wireless Packet Data Serving Node Configuration Guide*
- *Cisco IOS IP Mobility Configuration Guide*

## Benefits of Enhanced Object Tracking

- Increases the availability and speed of recovery of a network.
- Decreases network outages and their duration.
- Provides a scalable solution that allows other client processes such as VRRP and GLBP the ability to track objects individually or as a list of objects. Prior to the introduction of this functionality, the tracking process was embedded within HSRP.

# How to Configure Enhanced Object Tracking

- Tracking the Line-Protocol State of an Interface, page 6 (optional)
- Tracking the IP-Routing State of an Interface, page 8 (optional)
- Tracking IP-Route Reachability, page 10 (optional)
- Tracking the Threshold of IP-Route Metrics, page 12 (optional)
- Tracking the State of an IP SLAs Operation, page 14 (optional)

- Tracking the Reachability of an IP SLAs IP Host, page 15 (optional)
- Configuring a Tracked List and Boolean Expression, page 17 (optional)
- Configuring a Tracked List and Threshold Weight, page 19 (optional)
- Configuring a Tracked List and Threshold Percentage, page 20 (optional)
- Configuring the Track List Defaults, page 22 (optional)
- Configuring Tracking for Mobile IP Applications, page 23 (optional)

# Tracking the Line-Protocol State of an Interface

Perform this task to track the line-protocol state of an interface.

Tracking the IP-routing state of an interface using the **track interface ip routing** command can be more useful in some situations than just tracking the line-protocol state using the **track interface line-protocol** command, especially on interfaces for which IP addresses are negotiated. See the "Tracking the IP-Routing State of an Interface" section for more information.

You can optionally configure EOT to consider the carrier-delay timer when tracking the line-protocol state of an interface by using the **carrier-delay** command in tracking configuration mode.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track timer interface** *seconds* | **msec** *milliseconds*}
4. **track** *object-number* **interface** *type number* **line-protocol**
5. **carrier-delay**
6. **delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*}
7. **end**
8. **show track** *object-number*

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | **track timer interface** *seconds* \| **msec** *milliseconds*}<br><br>**Example:**<br>`Router(config)# track timer interface 5` | (Optional) Specifies the interval in which the tracking process polls the tracked object.<br>• The default interval that the tracking process polls interface objects is 1 second.<br>**Note** All polling frequencies can be configured down to 500 milliseconds, overriding the minimum 1 second interval configured previously using the **msec** keyword and *milliseconds* argument. |
| **Step 4** | **track** *object-number* **interface** *type number* **line-protocol**<br><br>**Example:**<br>`Router(config)# track 3 interface ethernet 0/1 line-protocol` | Tracks the line-protocol state of an interface and enters tracking configuration mode. |
| **Step 5** | **carrier-delay**<br><br>**Example:**<br>`Router(config-track)# carrier-delay` | (Optional) Enables EOT to consider the carrier-delay timer when tracking the status of an interface. |
| **Step 6** | **delay** {**up** *seconds* [**down** *seconds*] \| [**up** *seconds*] **down** *seconds*}<br><br>**Example:**<br>`Router(config-track)# delay up 30` | (Optional) Specifies a period of time (in seconds) to delay communicating state changes of a tracked object. |
| **Step 7** | **end**<br><br>**Example:**<br>`Router(config-track)# end` | Exits to privileged EXEC mode. |
| **Step 8** | **show track** *object-number*<br><br>**Example:**<br>`Router# show track 3` | (Optional) Displays tracking information.<br>• Use this command to verify the configuration. See the display output in the "Examples" section. |

## Examples

The following example shows the state of the line protocol on an interface when it is tracked:

```
Router# show track 3

Track 3
   Interface Ethernet0/1 line-protocol
   Line protocol is Up
     1 change, last change 00:00:05
   Tracked by:
     HSRP Ethernet0/3 1
```

# Tracking the IP-Routing State of an Interface

Perform this task to track the IP-routing state of an interface. An IP-routing object is considered up when the following criteria exist:

- IP routing is enabled and active on the interface.
- The interface line-protocol state is up.
- The interface IP address is known. The IP address is configured or received through the Dynamic Host Configuration Protocol (DHCP) or IP Control Protocol (IPCP) negotiation.

Interface IP routing will go down when one of the following criteria exist:

- IP routing is disabled globally.
- The interface line-protocol state is down.
- The interface IP address is unknown. The IP address is not configured or received through DHCP or IPCP negotiation.

Tracking the IP-routing state of an interface using the **track interface ip routing** command can be more useful in some situations than just tracking the line-protocol state using the **track interface line-protocol** command, especially on interfaces for which IP addresses are negotiated. For example, on a serial interface that uses the Point-to-Point Protocol (PPP), the line protocol could be up (link control protocol [LCP] negotiated successfully), but IP could be down (IPCP negotiation failed).

The **track interface ip routing** command supports the tracking of an interface with an IP address acquired through any of the following methods:

- Conventional IP address configuration
- PPP/IPCP
- DHCP
- Unnumbered interface

You can optionally configure EOT to consider the carrier-delay timer when tracking the IP-routing state of an interface by using the **carrier-delay** command in tracking configuration mode.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track timer interface** {*seconds* | **msec** *milliseconds*}
4. **track** *object-number* **interface** *type number* **ip routing**
5. **carrier-delay**
6. **delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*}
7. **end**
8. **show track** *object-number*

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **track timer interface** {*seconds* \| **msec** *milliseconds*}<br><br>**Example:**<br>`Router(config)# track timer interface 5` | (Optional) Specifies the interval in which the tracking process polls the tracked object.<br>• The default interval that the tracking process polls interface objects is 1 second.<br>**Note** All polling frequencies can be configured down to 500 milliseconds, overriding the minimum 1 second interval configured previously using the **msec** keyword and *milliseconds* argument. |
| **Step 4** | **track** *object-number* **interface** *type number* **ip routing**<br><br>**Example:**<br>`Router(config)# track 1 interface ethernet 0/1 ip routing` | Tracks the IP-routing state of an interface and enters tracking configuration mode.<br>• IP-route tracking tracks an IP route in the routing table and the ability of an interface to route IP packets. |
| **Step 5** | **carrier-delay**<br><br>**Example:**<br>`Router(config-track)# carrier-delay` | (Optional) Enables EOT to consider the carrier-delay timer when tracking the status of an interface. |
| **Step 6** | **delay** {**up** *seconds* [**down** *seconds*] \| [**up** *seconds*] **down** *seconds*}<br><br>**Example:**<br>`Router(config-track)# delay up 30` | (Optional) Specifies a period of time (in seconds) to delay communicating state changes of a tracked object. |
| **Step 7** | **end**<br><br>**Example:**<br>`Router(config-track)# end` | Returns to privileged EXEC mode. |
| **Step 8** | **show track** *object-number*<br><br>**Example:**<br>`Router# show track 1` | Displays tracking information.<br>• Use this command to verify the configuration. See the display output in the "Examples" section. |

## Examples

The following example shows the state of IP routing on an interface when it is tracked:

```
Router# show track 1

Track 1
   Interface Ethernet0/1 ip routing
   IP routing is Up
     1 change, last change 00:01:08
   Tracked by:
     HSRP Ethernet0/3 1
```

# Tracking IP-Route Reachability

Perform this task to track the reachability of an IP route. A tracked object is considered up when a routing table entry exists for the route and the route is accessible.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track timer ip route** {*seconds* | **msec** *milliseconds*}
4. **track** *object-number* **ip route** *ip-address*/*prefix-length* **reachability**
5. **delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*}
6. **ip vrf** *vrf-name*
7. **end**
8. **show track** *object-number*

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **track timer ip route** {*seconds* \| **msec** *milliseconds*}<br><br>**Example:**<br>`Router(config)# track timer ip route 20` | (Optional) Specifies the interval in which the tracking process polls the tracked object.<br>• The default interval that the tracking process polls IP-route objects is 15 seconds.<br>**Note** All polling frequencies can be configured down to 500 milliseconds, overriding the minimum 1 second interval configured previously using the **msec** keyword and *milliseconds* argument. |
| **Step 4** | **track** *object-number* **ip route** *ip-address*/*prefix-length* **reachability**<br><br>**Example:**<br>`Router(config)# track 4 ip route 10.16.0.0/16 reachability` | Tracks the reachability of an IP route and enters tracking configuration mode. |
| **Step 5** | **delay** {**up** *seconds* [**down** *seconds*] \| [**up** *seconds*] **down** *seconds*}<br><br>**Example:**<br>`Router(config-track)# delay up 30` | (Optional) Specifies a period of time (in seconds) to delay communicating state changes of a tracked object. |
| **Step 6** | **ip vrf** *vrf-name*<br><br>**Example:**<br>`Router(config-track)# ip vrf VRF2` | (Optional) Configures a VPN routing and forwarding (VRF) table. |
| **Step 7** | **end**<br><br>**Example:**<br>`Router(config-track)# end` | Returns to privileged EXEC mode. |
| **Step 8** | **show track** *object-number*<br><br>**Example:**<br>`Router# show track 4` | (Optional) Displays tracking information.<br>• Use this command to verify the configuration. See the display output in the "Examples" section. |

## Examples

The following example shows the state of the reachability of an IP route when it is tracked:

```
Router# show track 4

Track 4
   IP route 10.16.0.0 255.255.0.0 reachability
   Reachability is Up (RIP)
     1 change, last change 00:02:04
   First-hop interface is Ethernet0/1
   Tracked by:
     HSRP Ethernet0/3 1
```

# Tracking the Threshold of IP-Route Metrics

Perform this task to track the threshold of IP route metrics.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track timer ip route** {*seconds* | **msec** *milliseconds*}
4. **track resolution ip route** {**eigrp** *resolution-value* | **isis** *resolution-value* | **ospf** *resolution-value* | **static** *resolution-value*}
5. **track** *object-number* **ip route** *ip-address*/*prefix-length* **metric threshold**
6. **delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*}
7. **ip vrf** *vrf-name*
8. **threshold metric** {**up** *number* [**down** *number*] | **down** *number* [**up** *number*]}
9. **end**
10. **show track** *object-number*

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | **track timer ip route** {*seconds* \| **msec** *milliseconds*}<br><br>**Example:**<br>`Router(config)# track timer ip route 20` | (Optional) Specifies the interval in which the tracking process polls the tracked object.<br>• The default interval that the tracking process polls IP-route objects is 15 seconds.<br>**Note** All polling frequencies can be configured down to 500 milliseconds, overriding the minimum 1 second interval configured previously using the **msec** keyword and *milliseconds* argument. |
| **Step 4** | **track resolution ip route** {**eigrp** *resolution-value* \| **isis** *resolution-value* \| **ospf** *resolution-value* \| **static** *resolution-value*}<br><br>**Example:**<br>`Router(config)# track resolution ip route eigrp 300` | (Optional) Specifies resolution parameters for a tracked object.<br>• Use this command to change the default metric resolution values. |
| **Step 5** | **track** *object-number* **ip route** *ip-address/prefix-length* **metric threshold**<br><br>**Example:**<br>`Router(config)# track 6 ip route 10.16.0.0/16 metric threshold` | Tracks the scaled metric value of an IP route to determine if it is above or below a threshold.<br>• The default down value is 255, which equates to an inaccessible route.<br>• The default up value is 254. |
| **Step 6** | **delay** {**up** *seconds* [**down** *seconds*] \| [**up** *seconds*] **down** *seconds*}<br><br>**Example:**<br>`Router(config-track)# delay up 30` | (Optional) Specifies a period of time (in seconds) to delay communicating state changes of a tracked object. |
| **Step 7** | **ip vrf** *vrf-name*<br><br>**Example:**<br>`Router(config-track)# ip vrf VRF1` | (Optional) Configures a VRF table. |
| **Step 8** | **threshold metric** {**up** *number* [**down** *number*] \| **down** *number* [**up** *number*]}<br><br>**Example:**<br>`Router(config-track)# threshold metric up 254 down 255` | (Optional) Sets a metric threshold other than the default value. |
| **Step 9** | **end**<br><br>**Example:**<br>`Router(config-track)# end` | Exits to privileged EXEC mode. |
| **Step 10** | **show track** *object-number*<br><br>**Example:**<br>`Router# show track 6` | (Optional) Displays tracking information.<br>• Use this command to verify the configuration. See the display output in the "Examples" section. |

## Examples

The following example shows the metric threshold of an IP route when it is tracked:

```
Router# show track 6

Track 6
   IP route 10.16.0.0 255.255.0.0 metric threshold
   Metric threshold is Up (RIP/6/102)
     1 change, last change 00:00:08
   Metric threshold down 255 up 254
   First-hop interface is Ethernet0/1
   Tracked by:
     HSRP Ethernet0/3 1
```

# Tracking the State of an IP SLAs Operation

Perform this task to track the state of an IP SLAs operation.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track** *object-number* **rtr** *operation-number* **state**
   or
   **track** *object-number* **ip sla** *operation-number* **state**
4. **delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*}
5. **end**
6. **show track** *object-number*

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| Step 3 | **Cisco IOS Releases Prior to 12.4(20)T, 12.2(33)SXI1, and 12.2(33)SRE**<br>**track** *object-number* **rtr** *operation-number* **state**<br><br>**Cisco IOS Release 12.4(20)T, 12.2(33)SXI1, 12.2(33)SRE, or Later Releases**<br>**track** *object-number* **ip sla** *operation-number* **state**<br><br>**Example: Cisco IOS Releases Prior to 12.4(20)T, 12.2(33)SXI1, and 12.2(33)SRE**<br>`Router(config)# track 2 rtr 4 state`<br><br>**Example: Cisco IOS Release 12.4(20)T, 12.2(33)SXI1, 12.2(33)SRE, or Later Releases**<br>`Router(config)# track 2 ip sla 4 state` | Tracks the state of an IP SLAs object and enters tracking configuration mode.<br><br>**Note** Effective with Cisco IOS Release 12.4(20)T, 12.2(33)SXI1, and 12.2(33)SRE the **track rtr** command was replaced by the **track ip sla** command. The **track rtr** command will be removed in a future release and is available only to aid the update of existing configurations to the **track ip sla** command. |
| Step 4 | **delay** {**up** *seconds* [**down** *seconds*] \| [**up** *seconds*] **down** *seconds*}<br><br>**Example:**<br>`Router(config-track)# delay up 60 down 30` | (Optional) Specifies a period of time (in seconds) to delay communicating state changes of a tracked object. |
| Step 5 | **end**<br><br>**Example:**<br>`Router(config-track)# end` | Exits to privileged EXEC mode. |
| Step 6 | **show track** *object-number*<br><br>**Example:**<br>`Router# show track 2` | (Optional) Displays tracking information.<br>• Use this command to verify the configuration. See the display output in the "Examples" section of this task. |

## Examples

The following example shows the state of the IP SLAs tracking:

```
Router# show track 2

Track 2
   IP SLA 1 state
   State is Down
     1 change, last change 00:00:47
   Latest operation return code: over threshold
   Latest RTT (millisecs) 4
   Tracked by:
     HSRP Ethernet0/1 3
```

# Tracking the Reachability of an IP SLAs IP Host

Perform this task to track the reachability of an IP host.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track** *object-number* **rtr** *operation-number* **reachability**

   or

   **track** *object-number* **ip sla** *operation-number* **reachability**
4. **delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*}
5. **end**
6. **show track** *object-number*

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **Cisco IOS Releases Prior to 12.4(20)T, 12.2(33)SXI1, and 12.2(33)SRE:**<br>**track** *object-number* **rtr** *operation-number* **reachability**<br><br>**Cisco IOS Release 12.4(20)T, 12.2(33)SXI1, 12.2(33)SRE, or Later Releases**<br>**track** *object-number* **ip sla** *operation-number* **reachability**<br><br>**Example: Cisco IOS Releases Prior to 12.4(20)T, 12.2(33)SRE, and 12.2(33)SXI1**<br>`Router(config)# track 2 rtr 4 reachability`<br><br>**Example: Cisco IOS Release 12.4(20)T, 12.2(33)SXI1, 12.2(33)SRE, or Later Releases**<br>`Router(config)# track 2 ip sla 4 reachability` | Tracks the reachability of an IP SLAs IP host and enters tracking configuration mode.<br>**Note** Effective with Cisco IOS Release 12.4(20)T, 12.2(33)SXI1, and 12.2(33)SRE, the **track rtr** command was replaced by the **track ip sla** command. The **track rtr** command will be removed in a future release and is available only to aid the update of existing configurations to the **track ip sla** command. |
| **Step 4** | **delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*}<br><br>**Example:**<br>`Router(config-track)# delay up 30 down 10` | (Optional) Specifies a period of time (in seconds) to delay communicating state changes of a tracked object. |

| | Command or Action | Purpose |
|---|---|---|
| Step 5 | **end**<br><br>**Example:**<br>`Router(config-track)# end` | Exits to privileged EXEC mode. |
| Step 6 | **show track** *object-number*<br><br>**Example:**<br>`Router# show track 3` | (Optional) Displays tracking information.<br>• Use this command to verify the configuration. See the display output in the "Examples" section of this task. |

## Examples

The following example shows whether the route is reachable:

```
Router# show track 3

Track 3
   IP SLA 1 reachability
   Reachability is Up
     1 change, last change 00:00:47
   Latest operation return code: over threshold
   Latest RTT (millisecs) 4
   Tracked by:
     HSRP Ethernet0/1 3
```

# Configuring a Tracked List and Boolean Expression

Perform this task to configure a tracked list of objects and 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.

You may also configure a tracked list state to be measured using a weight or percentage threshold. See "Configuring a Tracked List and Threshold Weight" section on page 19 and "Configuring a Tracked List and Threshold Percentage" section on page 20.

**Note** The "not" operator is specified for one or more objects and negates the state of the object.

## Prerequisites

An object must exist before it can be added to a tracked list.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track** *track-number* **list boolean** {**and** | **or**}
4. **object** *object-number* [**not**]

5. **delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*}
6. **end**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **track** *track-number* **list boolean** {**and** \| **or**}<br><br>**Example:**<br>`Router(config-track)# track 100 list boolean and` | Configures a tracked list object and enters tracking configuration mode. The keywords are as follows:<br>• **boolean**—Specifies that the state of the tracked 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. |
| **Step 4** | **object** *object-number* [**not**]<br><br>**Example:**<br>`Router(config-track)# object 3 not` | Specifies the object to be tracked. The *object-number* argument has a valid range from 1 to 1000. There is no default. The optional **not** keyword negates the state of the object.<br>**Note** The example means that when object 3 is up, the tracked list detects object 3 as down. |
| **Step 5** | **delay** {**up** *seconds* [**down** *seconds*] \| [**up** *seconds*] **down** *seconds*}<br><br>**Example:**<br>`Router(config-track)# delay up 3` | (Optional) Specifies a tracking delay in seconds between up and down states. |
| **Step 6** | **end**<br><br>**Example:**<br>`Router(config-track)# end` | Returns to privileged EXEC mode. |

# Configuring a Tracked List and Threshold Weight

Perform this task to configure a list of tracked objects, to specify that weight be used as the threshold, and to configure a weight for each of its objects. A tracked list contains one or more objects. Using a threshold weight, the state of each object is determined by comparing the total weight of all objects that are up against a threshold weight for each object.

You can also configure a tracked list state to be measured using a Boolean calculation or threshold percentage. See the "Configuring a Tracked List and Boolean Expression" section on page 17 and the "Configuring a Tracked List and Threshold Percentage" section on page 20.

## Prerequisites

An object must exist before it can be added to a tracked list.

## Restrictions

You cannot use the Boolean "not" operator in a weight or percentage threshold list.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track** *track-number* **list threshold weight**
4. **object** *object-number* [**weight** *weight-number*]
5. **threshold weight** {**up** *number* **down** *number* | **up** *number* | **down** *number*}
6. **delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*}
7. **end**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **track** *track-number* **list threshold weight**<br><br>**Example:**<br>`Router(config-track)# track 100 list threshold weight` | Configures a tracked list object and enters tracking configuration mode. The keywords are as follows:<br>• **threshold**—Specifies that the state of the tracked list is based on a threshold.<br>• **weight**—Specifies that the threshold is based on a specified weight. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 4** | **object** *object-number* [**weight** *weight-number*]<br><br>**Example:**<br>`Router(config-track)# object 3 weight 30` | Specifies the object to be tracked. The *object-number* argument has a valid range from 1 to 1000. There is no default. The optional **weight** keyword specifies a threshold weight for each object. |
| **Step 5** | **threshold weight** {**up** *number* **down** *number* \| **up** *number* \| **down** *number*}<br><br>**Example:**<br>`Router(config-track)# threshold weight up 30` | Specifies the threshold weight. The keywords and arguments are as follows:<br>• **up** *number*—Valid range is from 1 to 255.<br>• **down** *number*—Range depends upon what you select for the **up** keyword. For example, if you configure 25 for up, you will see a range from 0 to 24 for down. |
| **Step 6** | **delay** {**up** *seconds* [**down** *seconds*] \| [**up** *seconds*] **down** *seconds*}<br><br>**Example:**<br>`Router(config-track)# delay up 3` | (Optional) Specifies a tracking delay in seconds between up and down states. |
| **Step 7** | **end**<br><br>**Example:**<br>`Router(config-track)# end` | Returns to privileged EXEC mode. |

# Configuring a Tracked List and Threshold Percentage

Perform this task to configure a tracked list of objects, to specify that a percentage will be used as the threshold, and to specify a percentage for each object in the list. A tracked list contains one or more objects. Using the threshold percentage, the state of the list is determined by comparing the assigned percentage of each object to the list.

You may also configure a tracked list state to be measured using a Boolean calculation or threshold weight. See "Configuring a Tracked List and Boolean Expression" section on page 17 and "Configuring a Tracked List and Threshold Weight" section on page 19.

## Prerequisites

An object must exist before it can be added to a tracked list.

## Restrictions

You cannot use the Boolean "not" operator in a weight or percentage threshold list.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track** *track-number* **list threshold percentage**
4. **object** *object-number*

5. **threshold percentage** {**up** *number* [**down** *number*] | **down** *number* [**up** *number*]}
6. **delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*}
7. **end**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **track** *track-number* **list threshold percentage**<br><br>**Example:**<br>`Router(config-track)# track 100 list threshold percentage` | Configures a tracked list object and enters tracking configuration mode. The keywords are as follows:<br>• **threshold**—Specifies that the state of the tracked list is based on a threshold.<br>• **percentage**—Specifies that the threshold is based on a percentage. |
| **Step 4** | **object** *object-number*<br><br>**Example:**<br>`Router(config-track)# object 3` | Specifies the object to be tracked. The *object-number* argument has a valid range from 1 to 1000. There is no default. |
| **Step 5** | **threshold percentage** {**up** *number* [**down** *number*] \| **down** *number* [**up** *number*]}<br><br>**Example:**<br>`Router(config-track)# threshold percentage up 30` | Specifies the threshold percentage. The keywords and arguments are as follows:<br>• **up** *number*—Valid range is from 1 to 100.<br>• **down** *number*—Range depends upon what you have selected for the **up** keyword. For example, if you specify 25 as up, a range from 26 to 100 is displayed for the **down** keyword. |
| **Step 6** | **delay** {**up** *seconds* [**down** *seconds*] \| [**up** *seconds*] **down** *seconds*}<br><br>**Example:**<br>`Router(config-track)# delay up 3` | (Optional) Specifies a tracking delay in seconds between up and down states. |
| **Step 7** | **end**<br><br>**Example:**<br>`Router(config-track)# end` | Returns to privileged EXEC mode. |

# Configuring the Track List Defaults

Perform this task to configure a default delay value for a tracked list, a default object, and default threshold parameters for a tracked list.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track** *track-number*
4. **default** {**delay** | **object** *object-number* | **threshold percentage**}
5. **end**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **track** *track-number*<br><br>**Example:**<br>`Router(config)# track 3` | Enters tracking configuration mode. |
| **Step 4** | **default** {**delay** \| **object** *object-number* \| **threshold percentage**}<br><br>**Example:**<br>`Router(config-track)# default delay` | Specifies a default delay value for a tracked list, a default object, and default threshold parameters for a tracked list. The keywords and arguments are as follows:<br>• **delay**—Reverts to the default delay.<br>• **object** *object-number*—Specifies a default object for the track list. The valid range is from 1 to 1000.<br>• **threshold percentage**—Specifies a default threshold percentage. |
| **Step 5** | **end**<br><br>**Example:**<br>`Router(config-track)# end` | Returns to privileged EXEC mode. |

# Configuring Tracking for Mobile IP Applications

Perform this task to configure a tracked list of Mobile IP application objects.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track** *track-number* **application home-agent**
4. **exit**
5. **track** *track-number* **application pdsn**
6. **exit**
7. **track** *track-number* **application ggsn**
8. **end**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **track** *track-number* **application home-agent**<br><br>**Example:**<br>`Router(config)# track 100 application home-agent` | (Optional) Tracks the presence of Home Agent traffic on a router. |
| **Step 4** | **exit**<br><br>**Example:**<br>`Router(config-track)# exit` | Returns to global configuration mode. |
| **Step 5** | **track** *track-number* **application pdsn**<br><br>**Example:**<br>`Router(config)# track 100 application pdsn` | (Optional) Tracks the presence of PDSN traffic on a router. |
| **Step 6** | **exit**<br><br>**Example:**<br>`Router(config-track)# exit` | Returns to global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 7** | **track** *track-number* **application ggsn**<br><br>**Example:**<br>`Router(config)# track 100 application ggsn` | (Optional) Tracks the presence of GGSN traffic on a router. |
| **Step 8** | **end**<br><br>**Example:**<br>`Router(config)# end` | Returns to privileged EXEC mode. |

# Configuration Examples for Enhanced Object Tracking

This section provides the following configuration examples:



## Example: Interface Line Protocol

The following example is very similar to the IP-routing example. Instead, the tracking process is configured to track the line-protocol state of serial interface 1/0. HSRP on Ethernet interface 0/0 then registers with the tracking process to be informed of any changes to the line-protocol state of serial interface 1/0. If the line protocol on serial interface 1/0 goes down, the priority of the HSRP group is reduced by 10.

### Router A Configuration

```
RouterA(config)# track 100 interface serial1/0 line-protocol
RouterA(config-track)# exit
RouterA(config)# interface Ethernet0/0
RouterA(config-if)# ip address 10.1.0.21 255.255.0.0
RouterA(config-if)# standby 1 preempt
RouterA(config-if)# standby 1 ip 10.1.0.1
RouterA(config-if)# standby 1 priority 110
RouterA(config-if)# standby 1 track 100 decrement 10
```

### Router B Configuration

```
RouterB(config)# track 100 interface serial1/0 line-protocol
RouterB(config-track)# exit
RouterB(config)# interface Ethernet0/0
RouterB(config-if)# ip address 10.1.0.22 255.255.0.0
```

```
RouterB(config-if)# standby 1 preempt
RouterB(config-if)# standby 1 ip 10.1.0.1
RouterB(config-if)# standby 1 priority 105
RouterB(config-if)# standby 1 track 100 decrement 10
```

# Example: Interface IP Routing

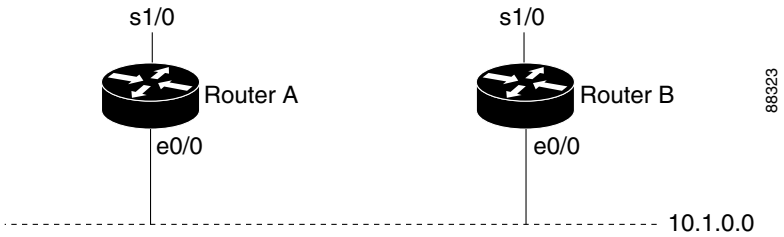
In the following example, the tracking process is configured to track the IP-routing capability of serial interface 1/0. HSRP on Ethernet interface 0/0 then registers with the tracking process to be informed of any changes to the IP-routing state of serial interface 1/0. If the IP-routing state on serial interface 1/0 goes down, the priority of the HSRP group is reduced by 10.

In the following example, EOT is configured to take the carrier-delay timer into consideration when tracking the state of serial interface 1/0.

If both serial interfaces are operational, Router A will be the HSRP active router because it has the higher priority. However, if IP on serial interface 1/0 in Router A fails, the HSRP group priority will be reduced and Router B will take over as the active router, thus maintaining a default virtual gateway service to hosts on the 10.1.0.0 subnet.

See Figure 1 for a sample topology.

***Figure 1 Topology for IP-Routing Support***




### Router A Configuration

```
RouterA(config)# track 100 interface serial1/0 ip routing
RouterA(config-track)# carrier-delay
RouterA(config-track)# exit
RouterA(config)# interface Ethernet0/0
RouterA(config-if)# ip address 10.1.0.21 255.255.0.0
RouterA(config-if)# standby 1 preempt
RouterA(config-if)# standby 1 ip 10.1.0.1
RouterA(config-if)# standby 1 priority 110
RouterA(config-if)# standby 1 track 100 decrement 10
```

### Router B Configuration

```
RouterB(config)# track 100 interface serial1/0 ip routing
RouterB(config-track)# carrier-delay
RouterB(config-track)# exit
RouterB(config)# interface Ethernet0/0
RouterB(config-if)# ip address 10.1.0.22 255.255.0.0
RouterB(config-if)# standby 1 preempt
RouterB(config-if)# standby 1 ip 10.1.0.1
RouterB(config-if)# standby 1 priority 105
RouterB(config-if)# standby 1 track 100 decrement 10
```

# Example: IP-Route Reachability

In the following example, the tracking process is configured to track the reachability of IP route 10.2.2.0/24:

### Router A Configuration

```
RouterA(config)# track 100 ip route 10.2.2.0/24 reachability
RouterA(config-track)# exit
RouterA(config)# interface Ethernet0/0
RouterA(config-if)# ip address 10.1.1.21 255.255.255.0
RouterA(config-if)# standby 1 preempt
RouterA(config-if)# standby 1 ip 10.1.1.1
RouterA(config-if)# standby 1 priority 110
RouterA(config-if)# standby 1 track 100 decrement 10
```

### Router B Configuration

```
RouterB(config)#track 100 ip route 10.2.2.0/24 reachability
RouterB(config-track)# exit
RouterB(config)# interface Ethernet0/0
RouterB(config-if)# ip address 10.1.1.22 255.255.255.0
RouterB(config-if)# standby 1 preempt
RouterB(config-if)# standby 1 ip 10.1.1.1
RouterB(config-if)# standby 1 priority 105
RouterB(config-if)# standby 1 track 100 decrement 10
```

# Example: IP-Route Threshold Metric

In the following example, the tracking process is configured to track the threshold metric of IP route 10.2.2.0/24:

### Router A Configuration

```
RouterA(config)# track 100 ip route 10.2.2.0/24 metric threshold
RouterA(config-track)# exit
RouterA(config)# interface Ethernet0/0
RouterA(config)# ip address 10.1.1.21 255.255.255.0
RouterA(config-if)# standby 1 preempt
RouterA(config-if)# standby 1 ip 10.1.1.1
RouterA(config-if)# standby 1 priority 110
RouterA(config-if)# standby 1 track 100 decrement 10
```

### Router B Configuration

```
RouterB(config)# track 100 ip route 10.2.2.0/24 metric threshold
RouterB(config-track)# exit
RouterB(config)# interface Ethernet0/0
RouterB(config-if)# ip address 10.1.1.22 255.255.255.0
RouterB(config-if)# standby 1 preempt
RouterB(config-if)# standby 1 ip 10.1.1.1
RouterB(config-if)# standby 1 priority 105
RouterB(config-if)# standby 1 track 100 decrement 10
```

# Example: IP SLAs IP Host Tracking

The following example shows how to configure IP host tracking for IP SLAs operation 1 in Cisco IOS releases prior to Cisco IOS Release 12.4(20)T, 12.2(33)SXI1, and 12.2(33)SRE:

```
Router(config)# ip sla 1
Router(config-ip-sla)# icmp-echo 10.51.12.4
Router(config-ip-sla-echo)# timeout 1000
Router(config-ip-sla-echo)# threshold 2
Router(config-ip-sla-echo)# frequency 3
Router(config-ip-sla-echo)# request-data-size 1400
Router(config-ip-sla-echo)# exit
Router(config)# ip sla schedule 1 start-time now life forever
Router(config-ip-sla)# track 2 rtr 1 state
Router(config-ip-sla)# exit
Router(config)# track 3 rtr 1 reachability
Router(config-track)# exit
Router(config)# interface ethernet0/1
Router(config-if)# ip address 10.21.0.4 255.255.0.0
Router(config-if)# no shutdown
Router(config-if)# standby 3 ip 10.21.0.10
Router(config-if)# standby 3 priority 120
Router(config-if)# standby 3 preempt
Router(config-if)# standby 3 track 2 decrement 10
Router(config-if)# standby 3 track 3 decrement 10
```

The following example shows how to configure IP host tracking for IP SLAs operation 1 in Cisco IOS Release 12.4(20)T, 12.2(33)SXI1, 12.2(33)SRE, and later releases:

```
Router(config)# ip sla 1
Router(config-ip-sla)# icmp-echo 10.51.12.4
Router(config-ip-sla-echo)# threshold 2
Router(config-ip-sla-echo)# timeout 1000
Router(config-ip-sla-echo)# frequency 3
Router(config-ip-sla-echo)# request-data-size 1400
Router(config-ip-sla-echo)# exit
Router(config)# ip sla schedule 1 start-time now life forever
Router(config)# track 2 ip sla 1 state
Router(config-track)# exit
Router(config)# track 3 ip sla 1 reachability
Router(config-track)# exit
Router(config)# interface ethernet0/1
Router(config-if)# ip address 10.21.0.4 255.255.0.0
Router(config-if)# no shutdown
Router(config-if)# standby 3 ip 10.21.0.10
Router(config-if)# standby 3 priority 120
Router(config-if)# standby 3 preempt
Router(config-if)# standby 3 track 2 decrement 10
Router(config-if)# standby 3 track 3 decrement 10
```

# Example: Boolean Expression for a Tracked List

In the following example, a track list object is configured to track two serial interfaces when both serial interfaces are up and when either serial interface is down:

```
Router(config)# track 1 interface serial2/0 line-protocol
Router(config-track)# track 2 interface serial2/1 line-protocol
Router(config-track)# exit
Router(config)# track 100 list boolean and
```

```
Router(config-track)# object 1
Router(config-track)# object 2
```

In the following example, a track list object is configured to track two serial interfaces when either serial interface is up and when both serial interfaces are down:

```
Router(config)# track 1 interface serial2/0 line-protocol
Router(config)# track 2 interface serial2/1 line-protocol
Router(config-track)# exit
Router(config)# track 101 list boolean or
Router(config-track)# object 1
Router(config-track)# object 2
```

The following configuration example shows that tracked list 4 has two objects and one object state is negated (if the list is up, the list detects that object 2 is down):

```
Router(config)# track 4 list boolean and
Router(config-track)# object 1
Router(config-track)# object 2 not
```

# Example: Threshold Weight for a Tracked List

In the following example, three serial interfaces in tracked list 100 are configured with a threshold weight of 20 each. The down threshold is configured to 0 and the up threshold is configured to 40:

```
Router(config)# track 1 interface serial2/0 line-protocol
Router(config-track)# track 2 interface serial2/1 line-protocol
Router(config-track)# track 3 interface serial2/2 line-protocol
Router(config-track)# exit
Router(config)# track 100 list threshold weight
Router(config-track)# object 1 weight 20
Router(config-track)# object 2 weight 20
Router(config-track)# object 3 weight 20
Router(config-track)# threshold weight down 0 up 40
```

The above example means that the track-list object goes down only when all three serial interfaces go down, and only comes up again when at least two serial interfaces are up (since 20+20 >= 40). The advantage of this configuration is that it prevents the track-list object from coming up if two interfaces are down and the third interface is flapping.

The following configuration example shows that if object 1 and object 2 are down, then track list 4 is up, because object 3 satisfies the up threshold value of up 30. But, if object 3 is down, both objects 1 and 2 need to be up in order to satisfy the threshold weight.

```
Router(config)# track 4 list threshold weight
Router(config-track)# object 1 weight 15
Router(config-track)# object 2 weight 20
Router(config-track)# object 3 weight 30
Router(config-track)# threshold weight up 30 down 10
```

This configuration may be useful to you if you have two small bandwidth connections (represented by object 1 and 2) and one large bandwidth connection (represented by object 3). Also the down 10 value means that once the tracked object is up, it will not go down until the threshold value is lower or equal to 10, which in this example means that all connections are down.

## Example: Threshold Percentage for a Tracked List

In the following example, four serial interfaces in track list 100 are configured for an up threshold percentage of 75. The track list is up when 75 percent of the serial interfaces are up and down when fewer than 75 percent of the serial interfaces are up.

```
Router(config)# track 1 interface serial2/0 line-protocol
Router(config-track)# track 2 interface serial2/1 line-protocol
Router(config-track)# track 3 interface serial2/2 line-protocol
Router(config-track)# track 4 interface serial2/3 line-protocol
Router(config-track)# exit
Router(config)# track 100 list threshold percentage
Router(config-track)# object 1
Router(config-track)# object 2
Router(config-track)# object 3
Router(config-track)# object 4
Router(config-track)# threshold percentage up 75
```

## Example: Mobile IP Application Tracking

The following example shows how to configure EOT to track Mobile IP, GGSN, and PDSN traffic on a router:

```
Router(config)# track 1 application home-agent
Router(config-track)# exit
Router(config)# track 2 application ggsn
Router(config-track)# exit
Router(config)# track 3 application pdsn
```

# Additional References

## Related Documents

| Related Topic | Document Title |
|---|---|
| Cisco IOS commands | *Cisco IOS Master Commands List, All Releases* |
| Embedded Event Manager | "Embedded Event Manager Overview" module |
| HSRP concepts and configuration tasks | "Configuring HSRP" module |
| GLBP concepts and configuration tasks | "Configuring GLBP" module |
| VRRP concepts and configuration tasks | "Configuring VRRP" module |
| GLBP, HSRP, and VRRP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples | *Cisco IOS IP Application Services Command Reference* |

# Standards

| Standards | Title |
|---|---|
| No new or modified standards are supported, and support for existing standards has not been modified. | — |

# MIBs

| MIBs | MIBs Link |
|---|---|
| No new or modified MIBs are supported, and support for existing MIBs has not been modified. | To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL:<br>http://www.cisco.com/go/mibs |

# RFCs

| RFCs | Title |
|---|---|
| No new or modified RFCs are supported, and support for existing RFCs has not been modified. | — |

# Technical Assistance

| Description | Link |
|---|---|
| The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password. | http://www.cisco.com/cisco/web/support/index.html |

# Feature Information for Enhanced Object Tracking

Table 3 lists the features in this module and provides links to specific configuration information.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

**Note** Table 3 lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

*Table 3 Feature Information for Enhanced Object Tracking*

| Feature Name | Releases | Feature Configuration Information |
|---|---|---|
| Enhanced Tracking Support | Cisco IOS XE 3.1.0SG<br>12.2(15)T<br>12.2(25)S<br>12.2(28)SB<br>12.2(33)SRA<br>12.2(33)SXH | The Enhanced Tracking Support feature separates the tracking mechanism from HSRP and creates a separate standalone tracking process that can be used by other Cisco IOS processes as well as HSRP. This feature allows tracking of other objects in addition to the interface line-protocol state.<br>The following sections provide information about this feature:<br>• Tracking the Line-Protocol State of an Interface, page 6<br>• Tracking the IP-Routing State of an Interface, page 8<br>• Tracking IP-Route Reachability, page 10<br>• Tracking the Threshold of IP-Route Metrics, page 12<br>The following commands were introduced or modified by this feature: **debug track**, **delay tracking**, **ip vrf**, **show track**, **standby track**, **threshold metric**, **track interface**, **track ip route**, **track timer**. |
| FHRP—Enhanced Object Tracking Integration with Embedded Event Manager | 12.2(33)SRB<br>12.2(33)SXI<br>12.4(2)T | EOT is now integrated with EEM to allow EEM to report on a status change of a tracked object and to allow EOT to track EEM objects.<br>The following section provides information about this feature:<br>• Enhanced Object Tracking and Embedded Event Manager, page 4<br>The following commands were introduced or modified by this feature: **action track read**, **action track set**, **default-state**, **event resource**, **event rf**, **event track**, **show track**, **track stub**. |

*Table 3 Feature Information for Enhanced Object Tracking (continued)*

| Feature Name | Releases | Feature Configuration Information |
|---|---|---|
| FHRP—Enhanced Object Tracking of IP SLAs Operations | Cisco IOS XE 3.1.0SG<br>12.2(25)S<br>12.2(27)SBC<br>12.2(33)SRA<br>12.2(33)SXH<br>12.3(4)T<br>15.0(1)S | This feature enables First Hop Redundancy Protocols (FHRPs) and other Enhanced Object Tracking (EOT) clients to track the output from IP SLAs objects and use the provided information to trigger an action.<br>The following section provides information about this feature:<br>• IP SLA Operation Tracking, page 3<br>• Tracking the State of an IP SLAs Operation, page 14<br>• Tracking the Reachability of an IP SLAs IP Host, page 15<br>• Example: IP SLAs IP Host Tracking, page 27<br>The following command was introduced by this feature: **track rtr**. |
| FHRP—Enhanced Object Tracking Support for Mobile IP | 12.4(11)T | The FHRP—Enhanced Object Tracking Support for Mobile IP feature provides new tracking objects needed by mobile wireless applications to track the presence of Home Agent, GGSN, or PDSN traffic on a router.<br>The following sections provide information about this feature:<br>• Enhanced Object Tracking for Mobile IP Applications, page 5<br>• Configuring Tracking for Mobile IP Applications, page 23<br>• Example: Mobile IP Application Tracking, page 29<br>The following command was introduced by this feature: **track application**. |
| FHRP—EOT Deprecation of **rtr** Keyword | 12.2(33)SRE<br>12.2(33)SXI1<br>12.4(20)T | This feature replaces the **track rtr** command with the **track ip sla** command.<br>The following sections provide information about this feature:<br>• IP SLA Operation Tracking, page 3<br>• Tracking the State of an IP SLAs Operation, page 14<br>• Tracking the Reachability of an IP SLAs IP Host, page 15<br>• Example: IP SLAs IP Host Tracking, page 27<br>The following command was introduced by this feature: **track ip sla**. |

*Table 3 Feature Information for Enhanced Object Tracking (continued)*

| Feature Name | Releases | Feature Configuration Information |
|---|---|---|
| FHRP—Object Tracking List | Cisco IOS XE 3.1.0SG<br>12.2(30)S<br>12.2(31)SB2<br>12.2(33)SRA<br>12.2(33)SXH<br>12.3(8)T<br>15.0(1)S | This feature enhances the tracking capabilities to enable the configuration of a combination of tracked objects in a list, and a flexible method of combining objects using Boolean logic.<br>The following sections provide information about this feature:<br>• Configuring a Tracked List and Boolean Expression, page 17<br>• Configuring a Tracked List and Threshold Weight, page 19<br>• Configuring a Tracked List and Threshold Percentage, page 20<br>• Configuring the Track List Defaults, page 22<br>The following commands were introduced or modified by this feature: **show track**, **threshold percentage**, **threshold weight**, **track list**, **track resolution**. |
| EOT Support for Carrier Delay | 12.4(9)T | The EOT Support for Carrier Delay feature enables Enhanced Object Tracking (EOT) to consider the carrier-delay timer when tracking the status of an interface.<br>The following sections provide information about this feature:<br>• EOT Support for Carrier Delay, page 4<br>• Tracking the Line-Protocol State of an Interface, page 6<br>• Tracking the IP-Routing State of an Interface, page 8<br>• Example: Interface IP Routing, page 25<br>The following commands were introduced or modified by this feature: **carrier-delay (tracking)**, **show track**. |

# Glossary

**DHCP**—Dynamic Host Configuration Protocol. DHCP is a protocol that delivers IP addresses and configuration information to network clients.

**GLBP**—Gateway Load Balancing Protocol. Provides automatic router backup for IP hosts that are configured with a single default gateway on an IEEE 802.3 LAN. Multiple first-hop routers on the LAN combine to offer a single virtual first-hop IP router while sharing the IP packet forwarding load. Other routers on the LAN may act as redundant (GLBP) routers that will become active if any of the existing forwarding routers fail.

**GGSN**—Gateway GPRS Support Node. A wireless gateway that allows mobile cell phone users to access the public data network (PDN) or specified private IP networks. The GGSN function is implemented on the Cisco routers.

**GPRS**—General Packet Radio Service. A 2.5G mobile communications technology that enables mobile wireless service providers to offer their mobile subscribers with packet-based data services over GSM networks.

**GSM network**—Global System for Mobile Communications network. A digital cellular technology that is used worldwide, predominantly in Europe and Asia. GSM is the world's leading standard in digital wireless communications.

**Home Agent**—A Home Agent is a router on the home network of the Mobile Node (MN) that maintains an association between the home IP address of the MN and its care-of address, which is the current location of the MN on a foreign or visited network. The HA redirects packets by tunneling them to the MN while it is away from the home network.

**HSRP**—Hot Standby Router Protocol. Provides high network availability and transparent network topology changes. HSRP creates a Hot Standby router group with a lead router that services all packets sent to the Hot Standby address. The lead router is monitored by other routers in the group, and if it fails, one of these standby routers inherits the lead position and the Hot Standby group address.

**IPCP**—IP Control Protocol. The protocol used to establish and configure IP over PPP.

**LCP**—Link Control Protocol. The protocol used to establish, configure, and test data-link connections for use by PPP.

**PDSN**—Packet Data Serving Node. The Cisco PDSN is a standards-compliant, wireless gateway that enables packet data services in a Code Division Multiplex Access (CDMA) environment. Acting as an access gateway, the Cisco PDSN provides simple IP and Mobile IP access, foreign-agent support, and packet transport for Virtual Private Networks (VPN).

**PPP**—Point-to-Point Protocol. Provides router-to-router and host-to-network connections over synchronous and asynchronous circuits. PPP is most commonly used for dial-up Internet access. Its features include address notification, authentication via CHAP or PAP, support for multiple protocols, and link monitoring.

**VRF**—VPN routing and forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a provider edge router.

**VRRP**—Virtual Router Redundancy Protocol. Eliminates the single point of failure inherent in the static default routed environment. VRRP specifies an election protocol that dynamically assigns responsibility for a virtual router to one of the VRRP routers on a LAN. The VRRP router that controls the IP addresses associated with a virtual router is called the master, and forwards packets sent to these IP addresses. The election process provides dynamic failover in the forwarding responsibility should the master become unavailable. Any of the virtual router IP addresses on a LAN can then be used as the default first-hop router by end hosts.