# WAN Monitoring

This reference outlines the monitoring features and diagnostic tools used to track WAN performance and status.



# WANMon

WANMon is a flexible solution that:

- addresses WAN link recovery requirements for physical networks and virtual links
- monitors WAN links for failure detection, and
- initiates link recovery actions on receipt of link failure triggers.

### Supported products and interfaces

WANMon supports the following networks and interfaces:

- Physical networks: 4G LTE and Ethernet (WAN port)
- Virtual links: Non-crypto map based IPSec tunnels (either legacy or FlexVPN); that is, any IPSec tunnel you configure as an interface.

## Built-in recovery actions

The following are the three levels of built-in recovery processes specific to the link type:

| Link Type | Recovery Actions | | |
|---|---|---|---|
| | Level 0 (Immediate) | Level 1 (Active) | Level 2 (Last-Resort) |
| 4G LTE | Clear interface, and then shut/no-shut | Module reload | System reload |
| Ethernet | Clear interface, and then shut/no-shut | No action taken | System reload |
| Tunnel | Shut/no-shut | No action taken | System reload |

Each level has two time-based thresholds based on which built-in recovery actions are taken. The following are the default settings for each level:

- *threshold* is the wait time in minutes after receipt of a link failure trigger to initiate the recovery action as set in the specified level.
- *mintime* is the frequency to perform the recovery action if the link remains down.

The built-in values are:

| Level | threshold | mintime | Description |
|---|---|---|---|
| Level 0 | 10 min | 10 min | Triggers Level 0 actions 10 minutes after the link went down. Repeat no more than every 10 minutes. |
| Level 1 | 60 min | 60 min | Triggers Level 1 actions 10 minutes after the link went down. Repeat no more than every 60 minutes. |
| Level 2 | 480 min | 60 min | Triggers Level 2 actions 480 minutes after the link went down. Repeat no more than every 60 minutes. |

**Note** If threshold values are specified as 0, no recovery actions are taken for that level. You can use this to avoid system reload (the built-in Level 2 recovery action) on receipt of a link failure trigger where other WAN links may be operational.

# Prerequisites

Provides the requirements for the WANMon module, specifically the inclusion of the **tm_wanmon.tcl** policy file in the IOS-XE image.

Ensure that the WANMon module is available. The WANMon module is included in the IOS-XE image as the **tm_wanmon.tcl** policy file.

# Guidelines and limitations

This section outlines the functional behavior of WANMon regarding IP address checking and event triggering, along with necessary precautions for live network environments.

These guidelines apply to WANMon operations:

- WANMon automatically performs IP address checking (no user configuration) as required for cellular interfaces.
- For all other interfaces, WANMon never performs IP address checking.
- WANMon indirectly triggers user-specified actions by generating an application event that link resetter applets monitor.
- If your network is live, ensure that you understand the potential impact of any command.

# Configure WANMon

You can enable WANMon on the router and assign WANMon support to specific interfaces. Optionally, you can override the built-in recovery actions, define custom recovery links, and define an event manager environment policy to set the track object value and disable IP address checking.

**Note** WANMon is disabled by default.

Follow these steps to configure WANMon.

**Procedure**

**Step 1** Use the **event manager policy tm_wanmon.tcl authorization bypass** command to enable the WANMon link recovery module.

**Example:**

```
Router(config)# event manager policy tm_wanmon.tcl authorization bypass
```

**Note**
Use the `authorization bypass` keywords to avoid authorization for CLIs invoked by this policy.

**Step 2** Use the **event manager environment wanmon_if_list** *instance* {*interface-name* {**ipsla** *instance*}} command to configure WANMon for the interfaces in your WAN.

**Example:**

```
Router(config)# event manager environment wanmon_if_list1 {cellular0/4/0 {ipsla 1}}
Router(config)# event manager environment wanmon_if_list2 {GigabitEthernet0/0/0 {ipsla 2}}
```

**Note**
- Any environment variable with the prefix `wanmon_if_list` constitutes an interface configuration.
- Multiple interfaces are allowed by specifying an instance.

- Be sure to specify the full interface name (for example, `cellular0/4/0` or `cellular0/5/0`).
- You can set the IP SLA `icmp-echo` trigger, if desired. Multiple IP SLA triggers are allowed by specifying an instance.
- WANMon only looks at the status of the SLA ID. Although `icmp-echo` is most common, you can use any other type of SLA probe (for example, `udp-echo`) instead, if needed.

**Step 3** (Optional) Use the **event manager environment wanmon_if_list***x* {*interface-name* {**recovery** *Level0* {*Level1*} *Level2*}} command to override the built-in thresholds.

**Step 4** (Optional) Use the **publish-event sub-system 798 type 2000 arg1** *interface-name* **arg2** *level* command to configure custom recovery actions using link resetter applets.

**Note**
- *interface-name* is the full interface name (for example, `cellular0/4/0` or `cellular0/5/0`).
- *level* is 0, 1, or 2 to match the desired link recovery action.

**Step 5** (Optional) Use the {**stub** *track-stub-id*} keyword to allow an event manager environment policy to set the track object value.

**Note**
WANMon can set a track-stub-object value to reflect the link state so that an external applet can track the stub object.

**Step 6** (Optional) Use the **event manager environment wanmon_if_list***x* {*interface-name* {**checkip** *instance*}} command to disable IP address checking.

---

The following example enables the WANMon link recovery module:

```
Router(config)# event manager policy tm_wanmon.tcl authorization bypass
```

The following examples configure cellular and Ethernet interfaces:

```
Router(config)# event manager environment wanmon_if_list1 {cellular0/4/0 {ipsla 1}}
Router(config)# event manager environment wanmon_if_list2 {GigabitEthernet0/0/0 {ipsla 2}}
```

The following example sets custom recovery thresholds:

```
Router(config)# event manager environment wanmon_if_list {cellular0/4/0 {recovery 20 {90
75} 600}
```

In this example:

- The Level 0 threshold is set to 20 minutes after the link failure trigger. Level 0 recovery actions are performed for the cellular interface and repeat indefinitely, no more than every 10 minutes (default).
- The Level 1 threshold is set to 90 minutes. Level 1 recovery actions are performed for the cellular interface and repeat no more frequently than every 75 minutes.
- The Level 2 threshold is set to 600 minutes (10 hours).

The following example sets the track-stub-object value to 21:

```
Router# configure terminal
Router(config)# track 21 stub-object
Router(config)# event manager environment wanmon_if_list {cellular0/4/0 {ipsla 1} {stub 21}
```

# Verify WANMon configuration

Follow these steps to verify your WANMon configuration.

**Procedure**

**Step 1** Use the **show event manager policy registered** command to display the WAN monitoring policy.

**Example:**

```
Router# show event manager policy registered
1    script    system  multiple   Off   Thu Jan 16 18:44:29 2014  tm_wanmon.tcl
```

**Step 2** Use the **show event manager environment** command to display the interface environment variables set during interface configuration.

**Example:**

```
Router# show event manager environment
1 wanmon_if_list   {cell0/4/0 {ipsla 1}}
```

# Configuration examples

The following examples are provided:

## WANMon cellular interface configuration example

This configuration example demonstrates the setup of WANMon for a cellular interface, including IP SLA tracking, ICMP echo configuration, and event manager policy settings.

```
track 1 ip sla 1
ip sla 1
 icmp-echo 172.27.166.250
 timeout 6000
 frequency 300
ip sla schedule 1 life forever start-time now
event manager environment wanmon_if_list {cellular0/4/0 {ipsla 1}}
event manager policy tm_wanmon.tcl authorization bypass
```

## Multiple WAN link monitoring example

This example demonstrates how to configure multiple WAN link monitoring using IP SLA operations with track objects and event manager policies. The configuration includes ICMP echo operations for two different target addresses with associated tracking objects and stub objects for failover scenarios.

```
track 1 ip sla 1
track 21 stub-object
ip sla 1
 icmp-echo 172.27.166.250
 timeout 6000
 frequency 300
ip sla schedule 1 life forever start-time now
track 2 ip sla 2
track 22 stub-object
ip sla 2
 icmp-echo 10.27.16.25
 timeout 6000
 frequency 300
ip sla schedule 2 life forever start-time now
event manager environment wanmon_if_list1 {cellular0/4/0 {ipsla 1} {stub 21}}
event manager policy tm_wanmon.tcl authorization bypass
```