WAN Monitoring

This chapter contains the following topics:

• Information About WANMon, on page 1
• Prerequisites, on page 2
• Guidelines and Limitations, on page 2
• Configuring WANMon, on page 2
• Verifying WANMon Configuration, on page 4
• Configuration Examples, on page 5

Information About WANMon

WANMon is a flexible solution to address the WAN link recovery requirements for the following products 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.

You enable WANMon to monitor your WAN links and initiate link recovery actions on receipt of link failure triggers.

Built-in Recovery Actions

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

<table>
<thead>
<tr>
<th>Link Type</th>
<th>Recovery Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Level 0 (Immediate)</td>
</tr>
<tr>
<td>4G LTE</td>
<td>Clear interface, and then shut/no-shut</td>
</tr>
<tr>
<td>Ethernet</td>
<td>Clear interface, and then shut/no-shut</td>
</tr>
<tr>
<td>Tunnel</td>
<td>Shut/no-shut</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>Level</th>
<th>threshold</th>
<th>mintime</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 0</td>
<td>10 min</td>
<td>10 min</td>
<td>Triggers Level 0 actions 10 minutes after the link went down. Repeat no more than every 10 minutes.</td>
</tr>
<tr>
<td>Level 1</td>
<td>60 min</td>
<td>60 min</td>
<td>Triggers Level 1 actions 10 minutes after the link went down. Repeat no more than every 60 minutes.</td>
</tr>
<tr>
<td>Level 2</td>
<td>480 min</td>
<td>60 min</td>
<td>Triggers Level 2 actions 480 minutes after the link went down. Repeat no more than every 60 minutes.</td>
</tr>
</tbody>
</table>

*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

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

- 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.

### Configuring 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. WANMon is disabled by default.

**SUMMARY STEPS**

1. event manager policy `tm_wanmon.tcl` authorization bypass
2. event manager environment `wanmon_if_list <instance> {interface name {ipsla <instance>}}`
3. event manager environment `wanmon_if_lists {interface name {recovery Level0 {Level1 Level2}}}`
4. publish-event sub-system 798 type 2000 arg1 `<interface name>` arg2 `<level>`
5. {stub <track-stub-id> }
6. event manager environment `wanmon_if_lists {<interface name> {checkip <instance>}}`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables the WANMon link recovery module. Use authorization bypass to avoid authorization for CLIs invoked by this policy.</td>
</tr>
<tr>
<td>event manager policy <code>tm_wanmon.tcl</code> authorization bypass</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Configures WANMon for the interfaces in your WAN, and indicates that this is an interface configuration command.</td>
</tr>
<tr>
<td>event manager environment <code>wanmon_if_list &lt;instance&gt; {interface name {ipsla &lt;instance&gt;}}</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>(Optional) Overrides the built-in thresholds.</td>
</tr>
<tr>
<td>event manager environment <code>wanmon_if_lists {interface name {recovery Level0 {Level1 Level2}}}</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>(Optional) Configures custom recovery actions using link resetter applets.</td>
</tr>
<tr>
<td>publish-event sub-system 798 type 2000 arg1 <code>&lt;interface name&gt;</code> arg2 <code>&lt;level&gt;</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>(Optional) Allows an event manager environment policy to set the track object value. WANMon can set a</td>
</tr>
<tr>
<td>{stub <code>&lt;track-stub-id&gt;</code> }</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td>track-stub-object value to reflect the link state so that an external applet can track the stub object.</td>
<td></td>
</tr>
</tbody>
</table>

**Step 6**

```bash
(event manager environment wanmon_if_listx {<interface name> {checkip <instance>}})
```

(Optional) Disables IP address checking.

**What to do next**

**EXAMPLES**

```bash
event manager policy tm_wanmon.tcl authorization bypass
```

The following examples are Event Manager commands to configure cellular and Ethernet interfaces:

```bash
event manager environment wanmon_if_list1 {cellular0/1/0 {ipsla 1}}
event manager environment wanmon_if_list2 {GigabitEthernet0/0/0 {ipsla 2}}
```

This example sets custom recovery thresholds:

```bash
event manager environment wanmon_if_list {cellular0/1/0 {recovery 20 {90 75} 600}}
```

where:

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

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

```bash
conf t
track 21 stub-object
(event manager environment wanmon_if_list {cellular0/1/0 {ipsla 1} {stub 21}}
```

**Verifying WANMon Configuration**

Use the following steps to verify your WANMon configuration.

**SUMMARY STEPS**

1. show event manager policy registered
2. show event manager environment
DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> show event manager policy registered</td>
<td>Displays the WAN monitoring policy.</td>
</tr>
<tr>
<td><strong>Step 2</strong> show event manager environment</td>
<td>Displays the interface environment variables set during interface configuration.</td>
</tr>
</tbody>
</table>

What to do next

**EXAMPLE**

```
show event manager policy registered
1  script  system  multiple  Off  Thu Jan 16 18:44:29 2014  tm_wanmon.tcl
show event manager environment
1  wanmon_if_list  {cell0/1/0 {ipsla 1}}
```

Configuration Examples

The following examples are provided:

**WANMon Cellular Interface Configuration Example**

```
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  {cell0/1/0 {ipsla 1}}
  event manager policy tm_wanmon.tcl  authorization  bypass
```

**Multiple WAN Link Monitoring Example**

```
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_list  {cell0/1/0 {ipsla 1} {stub 21}}
  event manager policy tm_wanmon.tcl  authorization  bypass
```