



APPENDIX **F**

VNE Persistency Mechanism

This appendix describes the VNE persistency mechanism in Cisco ANA.



Note

Changes to the registry should only be carried out with the support of Cisco Professional Services.

Introducing Persistency

This section includes a description of:

- [Alarm Persistency](#)
- [Instrumentation Persistency](#)
- [Topology Persistency](#)

Persistency is the ability to store information in the unit for later use. This information is stored across unit, AVM and VNE restarts.

Instrumentation persistency is used mainly to:

- Shorten the starting time of VNEs for devices. By using the information from the local file system, the device's response time and network latency is eliminated, thus the VNE finishes modeling its first state very quickly.
- Provide information about the old state of the VNE in order to initiate alarms if the status has changed, if the VNE was unloaded for any reason. For example, a port-down alarm is initiated only if the port status was “up” and was changed to “down”. This will make sure that an alarm is not issued on ports which should be “down”. By maintaining information about the old state of the port, the system understands whether the current state is valid or not.
- Help lower the CPU load on the device when starting and there are many get commands that are generated. Also, when persistence data is loaded from the unit the traffic bandwidth between the unit and device is much lower than when the system is loaded ordinarily using “ordinary” device discovery and modulation.

Topology persistency is used mainly to:

- Create topology between devices on startup when the VNE is loaded, instead of performing the entire discovery process. Verification of the links is then performed.

Alarm persistency is used mainly to:

- Save information about the VNE components that send alarms. When a VNE sends an alarm it can be configured to save this information (that it has sent an alarm of type X). This information can then be used by the VNE components after restarts to verify whether it needs to send clearing alarms where changes have occurred in the device when the VNE was down.

VNE data persists:

- During runtime when a VNE polls data from a device, and it updates the files in the file system for changes in the device's response according to the persistency variables.
- The reading from these files is done once when the VNE starts. Every normal polling or refresh that takes place after this first time will read the data from the device itself and not from the files.

Alarm Persistency

Alarm persistency enables the system to clear alarms which relate to events which have occurred while the system was down. For example, a link-down alarm is generated, and then the system goes down. While the system is down a link-up event occurs in the network, but the system is down and does not monitor the network. When the system goes up, the alarm will be cleared because the system remembers that a link-down alarm exists and needs to be cleared by sending a corresponding alarm.

Persisting events are held in the AlarmPersistencyManager. Each VNE contains an AlarmPersistencyManager object. Alarms are added and removed from the AlarmPersistencyManager object in order to maintain the status of an event, whether it exists in the repository or not, namely, whether an up alarm has been generated, or whether a clearing alarm has been generated. Two copies of alarm persistency information are maintained, one in the memory, and the other on disk.

At startup, the AlarmPersistencyManager retrieves the events persisting for the containing VNE.

Event data in the files is updated:

- At shutdown
or
- After a change when a new event is added or removed
or
- After a specific interval of time has passed. This prevents data from being rewritten to the persistency file when a stream of events is added, or removed during a short period of time because the data is saved only after the specified period of time has elapsed.

Initialization

The AlarmPersistencyManager reads the following configurable items:

- Enabled—The mechanism enabled for this VNE.
- Writing delay—The interval between the arrival of a new event or a removal of an existing event, and the writing activity of the persistency file.
- Maximum age—How long an event remains in the persistency files before it becomes obsolete.



Note This only applies when trying to retrieve data from the persistency files.

Retrieving Events

At startup, each VNE calls its AlarmPersistencyManager to load the persisting events.

If the file does not exist or is corrupt, no events are loaded. Faulty event objects are not be loaded. Events which have been in the file for longer then the configured amount of maximum age are not loaded. No age tests are held during ordinary runtime.

Storing Events

At shutdown, events are saved to the VNE's event persistency file as a precaution in case the events have not been saved.

Removing an Event

An event will be searched for and removed using the same information which was used to add it. The event is removed from the memory because an up alarm, for example, a link-up alarm, has been generated, and the persistency information is no longer required. After the removal, the AlarmPersistencyManager stores the events after a writing delay as specified in the registry.

Removing an Event and Clearing an Alarm

The AlarmPersistencyManager is able to search for and remove an event, and send a clearing alarm for this event if it is found that this information is no longer required as the alarm has been cleared.

After an event has been added or removed from the AlarmPersistencyManager, a delayed message is sent to the AlarmPersistencyManager which, on its arrival, triggers the events to be stored to the file.

Configuring Alarm Persistency

The user can define whether each sub-event has a persisting or unpersisting alarm. This can be defined as both or none. An example of the LDP neighbor loss alarm is displayed:

```
<key name="LDP neighbor loss">
  <entry name="default">event-persistency-application/templates/generic persistency
  event</entry>
  <key name="sub-types">
    <key name="LDP neighbor down">
      <entry name="alarm-persistency">persist</entry>
    </key>
    <key name="LDP neighbor up">
      <entry name="alarm-persistency">unpersist</entry>
    </key>
  </key>
</key>
```

This persistency information is stored in event-persistency-application.xml.

Alarm Persistency Default Configuration

The following alarms are configured to be persistent:

- BGP neighbour loss
- LDP neighbor loss
- MPLS interface removed
- ascend link down trap
- card down
- card out
- component unreachable
- cpu utilization
- discard packets
- dropped packets
- drops exceed limit
- duplicate ip on vpn
- l2tp peer not established
- l2tp sessions threshold
- link down
- lsp removed
- memory utilization
- port down
- rx utilization
- tx utilization

Instrumentation Persistency

The instrumentation layer persists the information that was collected from the device to the file system. When the VNE restarts, it uses this information to emulate the device's response and thus the VNE can be modeled according to its last persistent state. The next polling instance is performed against the real device.

Configuring Instrumentation Persistency

The following instrumentation parameters can be configured:

- `persistencydir`
- `persistencylevel`
- `persistencystorageenabled`
- `persistencystorageinterval`
- `persistencytimeout`

persistencydir

This is the directory in which the persistency information is saved on the local file system. This is a relative path.

- Allowed values—A string representing the relative directory in the file system.
- Default value—instrumentor-persistency.

persistencylevel

This is the level of persistency to be used. The allowed values are:

- Full—Means it will persist. This is the default value.
- Off—Means it will not persist.
- Partial—This should not be used.

These values may be on certain commands to make sure some are persistent and some not.

**Note**

This is a command level parameter, meaning you can decide that one command is persisted using full and another is not using off.

persistencystorageenabled

This defines whether to enable the whole mechanism or not.

- Allowed values—True or false.
- Default value—True.

persistencystorageinterval

This is the interval in milliseconds, for which the data to be persisted is accumulated and then written to the persistent storage in bulk, in order to use less IO operations.

- Allowed values—Within the user's discretion.

**Note**

Small intervals cause more IO operations on the local file system. Very long intervals means that the information that is stored is less up-to-date.

- Default value—600000 (10 minutes)

persistencytimeout

If the persistency mechanism is enabled when the instrumentation layer starts, it loads all the data from the files, and this data can be used as data for the commands only for the first time they are executed. Some commands can be used for the first time, long after all the other commands have finished multiple cycles, for example, commands which run only when the status on the device has changed. This initial data is marked as obsolete after the persistency timeout has passed, and commands after this time, even if they are run for the first time, will run directly on the device. The time is defined in milliseconds.

- Allowed values—Within the user's discretion, should however usually be at least one minute.

**Note**

A small value causes the instrumentation layer to ignore the persistent data. A large value causes old data to be retrieved long after the VNE has finished loading.

- Default value—60000 (1 minute)

Topology Persistency

Cisco ANA supports persistency for Layer 1 topological connections. Layer 1 topology supports one connection per DC, so the physical topology reflects a single port connected by a single link.

The following topologies are persisted:

- Layer 1 counter-based topologies.
- Static topologies.
- Path-based topologies for B-STDx, GX and CBX.

Static topology, which identifies physical links configured by the user, is persisted once a user configures the static link between the two entities. This link is then stored in the registry in the AVM key that contains the specific VNE registrations.

For other topology, every time a link is created the persistency mechanism writes the link to this file. When a link is disconnected, the file representing the link is removed.

Configuring Topology Persistency

Physical topology persistency can be enabled or disabled in the registry. Topology has a registry entry entitled Persistency:

- The entry can be defined as true or false value.
- In order to enable topology persistency the value should be defined as true.
- The default value is true.

**Note**

Topology persistency assumes that the XID (unique device component ID) will be persistable. For example, the port XID should remain the same XID after the device reboots or after the VNE reboots. This is not dependent on whether the ifIndex is changed from time to time.