CHAPTER 26

VNE Persistency Mechanism

Persistency is the ability to store information in the unit for later use. These topics describe the VNE persistency mechanism in Prime Network:

Note These topics describe some of the persistency registry settings. Changes to the registry should be performed only with the support of Cisco. For details, contact your Cisco account representative.

Persistency Overview

Persistency information is stored across unit, AVM, and VNE restarts. VNE data persists during runtime when a VNE polls data from a device, and the VNE updates the files in the file system for changes in the device's response according to the persistency variables. When a VNE is started or restarted, the persistency information is read from these files once. Every normal polling or refresh that takes place after the first time will read the data from the device itself and not from the files.

VNE data persistency is lost in the following scenarios (but alarm persistency is saved):

- A user manually moves the VNE to another AVM, or moves the parent AVM to another unit.
- A unit server high availability event occurs, causing a unit to switch over to the standby unit.
- The device the VNE models is reconfigured (for example, a new sysOID or software version change).

The upgrade mechanism automatically clears all persistency files on Prime Network gateways and units. This option does not clear the alarm history that is stored in the Prime Network database.

Instrumentation Persistency

Instrumentation persistency is used mainly to:

- Shorten the starting time of VNEs for devices. When the information from the local file system is used, the device's response time and network latency are eliminated; thus the VNE finishes modeling its first state very quickly.
- Provide information about the old state of the VNE, to initiate alarms if the status has changed while the VNE was unloaded. For example, a Port Down alarm is initiated only if the port status was up and changed to down. This ensures 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 or not the current state is valid.
- Help lower the CPU load on the device while starting when many polling commands are generated. Also, when persistence data is loaded from the unit, traffic bandwidth between the unit and device is much lower than when the system is loaded using "ordinary" device discovery and modeling.

For more information, see Instrumentation Persistency, page 26-6.

Topology Persistency

Topology persistency creates topology between devices on startup when the VNE is loaded, instead of performing the entire discovery process. Verification of the links is then performed. For more information, see Topology Persistency, page 26-7.

Alarm Persistency

Alarm persistency saves information about the VNE components that send alarms. When a VNE sends an alarm, the VNE can 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 the VNE needs to send clearing alarms where changes have occurred in the device when the VNE was down. For more information, see Alarm Persistency, page 26-2.

Alarm Persistency

Alarm persistency enables the system to clear alarms that relate to events that 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 because the system is down, it does not monitor the network. When the system goes up, the alarm is cleared because the system remembers that a Link Down alarm exists, and the system needs to clear it by sending a corresponding alarm.

Persisting events are held in the AlarmPersistencyManager. Each VNE contains an AlarmPersistencyManager object. Alarms are added to and removed from the AlarmPersistencyManager object in order to maintain the status of an event, whether it exists in the repository or not; that is, whether an up alarm or 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 persisted for the containing VNE.

Event data in the files is updated at the following times:

- At shutdown.
- After a change, when an event is added or removed.
- 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

Alarm persistency is controlled by settings in the registry. Global alarm persistency information is stored in agentdefaults.xml. The major settings are listed in Table 26-1. The settings for these configurable items only apply when trying to retrieve data from the persistency files. Individual event persistency information is described in Configuring Alarm Persistency for a Specific Event, page 26-4.

Note All changes to the registry should only be carried out with the support of Cisco. For details, contact your Cisco account representative.

Table 26-1 Default Settings for Alarm Persistency

Registry Entry	Description	Default Value
enabled	Enabled the persistency mechanism for this VNE.	true
writing-delay	Interval (in milliseconds) between the arrival of a new event or the removal of an existing event, and the writing activity of the persistency file.	300000 (5 minutes)
max-alarm-age-in-days	How many days an event remains in a persistency file before it becomes obsolete.	7

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 loaded. Events which have been in the file for longer than the configured 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 already been saved.

Removing an Event

An event is searched for and removed using the same information which was used to add it. The event is removed from memory because a clearing event (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 the event, if it is found that this information is no longer required because the alarm has been cleared.

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

Configuring Alarm Persistency for a Specific Event

Alarm persistency can be configured per event using the setting described in Table 26-2. Event-specific persistency information is stored in event-persistency-application.xml.

Note All changes to the registry should only be carried out with the support of Cisco. For details, contact your Cisco account representative.

Table 26-2 Registry Setting for Alarm Persistency for a Specific Event

Registry Entry	Description	Default Value
alarm-persistency	Enable persistency for a specific event.	See Alarm Persistency Default Configuration, page 26-4

In the following LDP Neighbor Loss alarm, the LDP Neighbor Down event marks the alarm as present in the system (persisted), and the LDP Neighbor up event is used to clear the alarm from persistency (unpersist):

```
<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>
```

Alarm Persistency Default Configuration

The following alarms are configured to be persistent.

Table 26-3 Persisted Alarms

all ip interfaces down	ds3path port down due to admin	link down on unreachable
ascend link down trap	ds3path port down due to card	link overutilized
bfd connectivity down	ds3path port down due to oper	log archive disabled
bfd neighbour loss	ds3path port down due to upper layer down	low priority member down
bgp link down due to admin	ds3path port flapping	lsp removed
bgp link down due to oper	dual stack IP removed	medium priority member down
bgp link down vrf due to admin	duplicate ip on vpn found	memory overutilized
bgp link down vrf due to oper	dwdm controller down	mlppp admin down
bgp neighbour loss due to admin	dwdm g709 status down	mlppp oper down
bgp neighbour loss due to oper	efp admin down	MPLS TE FRR state changed to active
bgp-neighbor-loss-vrf-due-to-admin	efp down due to error disabled	MPLS interface removed

Table 26-3 Persisted Alarms (continued)

bgp-neighbor-loss-vrf-due-to-oper	efp oper down	ospf neighbor down
bridgeilan ac clear	envmon condition syslog	pim interface down syslog
bridgeilan ac shutdown	envmon fan syslog	pim neighbor loss syslog
bridgeilan bridge-domain clear	envmon powersupply syslog	port down due to admin
bridgeilan bridge-domain shutdown	envmon temperature syslog	port down due to card out
bridgeilan pseudowire shutdown	fabric hardware syslog	port down due to card down
bridgeilan pseudowire clear	flash card removed syslog	port down due to oper
card down	GRE tunnel down	port down due to upper layer down
card down syslog	high priority member down	port flapping
card out	ima admin down	rx dormant
cpu overutilized	ima oper down	rx overutilized
device unsupported	interface status down GRE tunnel	sonetpath link down
discard packets	interface status down connection	sonetpath link down due to admin down
dropped packets	interface status down non connection	sonetpath link down due to card
ds0 bundle admin down	keepalive not set	sonetpath link down due to oper down
ds0 bundle oper down	l2tp peer not established	sonetpath link down on unreachable
ds1path link down	l2tp sessions count exceeds max threshold	sonetpath port down due to admin
ds1path link down due to admin down	lag admin down	sonetpath port down due to card
ds1path link down due to card	lag oper down	sonetpath port down due to oper
ds1path link down due to oper down	lag link admin down	sonetpath port flapping
ds1path link down on unreachable	lag link down on unreachable	stop flapping non-cleared
ds1path port down due to admin	lag link oper down	sub card down
ds1path port down due to card	layer 2 aggregation admin down	sub card out
ds1path port down due to oper	layer 2 aggregation oper down	sub-interface admin down
ds1path port down due to upper layer downb	layer 2 tunnel down	sub-interface oper down
ds1path port flapping	LDP neighbor down	tx dormant
ds3path link down	link down	tx overutilized
ds3path link down due to admin down	link down due to admin down	vsi admin down
ds3path link down due to card	link down due to card	vsi oper down
ds3path link down due to oper down	keepalive not set	
ds3path link down on unreachable	link down due to oper down	

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.

The registry entries that control instrumentation persistency are provided in Table 26-4.

Note All changes to the registry should only be carried out with the support of Cisco. For details, contact your Cisco account representative.

Table 26-4 Registry Settings for Instrumentation Persistency

Registry Entry	Description	Default Value
persistencydir	Specifies the directory in which persistency information is saved on the local file system. This is a relative path. Allowed values are a string that represents the relative directory in the file system.	instrumentor-persistency
persistencylevel	Controls the level of persistency to be used. The allowed values are Full (persisted) or Off (not persisted). These values can be used for certain commands to make sure some are persisted and some are not. **Note** If a compound command contains both Full and Off persistency levels, Prime Network will use the full level for all commands.	Full
persistencystorageenabled	Controls whether the whole storage mechanism is enabled.	true
persistencystorageinterval	Interval (in milliseconds) for which the data to be persisted is accumulated and then written to the persistent storage in bulk. Files are only updated if they have changed. The default value (20 minutes) is a compromise between small intervals (which cause more I/O operations in the local file system) and long intervals (which result in stored information not being up-to-date).	1200000 (20 minutes)
persistencytimeout	Timeout period (in milliseconds) at which initial data is marked as obsolete; all subsequent commands will run directly on the device. If the persistency mechanism is enabled when the instrumentation layer starts, it loads all the data from the files. This data can be used for the commands only the first time they are executed. Some commands can be used for the first time, long after other commands have finished multiple cycles; for example, commands which run only when the status on the device has changed. The default value (1 minute) is a compromise between a small value (which can cause the instrumentation layer to ignore the persistent data) and a large value (which causes the data to be retrieved long after the VNE has finished loading). **Note** We recommend that this value be at least 600000 (1 minute).	600000 (1 minute)

Topology Persistency

Prime Network supports persistency for Layer 1 topological connections. Layer 1 topology supports one connection per Device Component (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.

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 topologies, 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.

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

Topology persistency is controlled by the setting listed in Table 26-5.

Note All changes to the registry should only be carried out with the support of Cisco. For details, contact your Cisco account representative.

Table 26-5 Registry Setting for Topology Persistency

Registry Entry	Description	Default Value
persistency	Enable physical topology persistency. **Note** We recommend that this entry remain enabled.	true