
Alarm Synchronization

In this section:

- [Overview](#)
- [Enabling Alarm Synchronization on the DSC](#)
- [Alarm State Synchronization](#)
- [Rest API Commands](#)

This section provides an overview of alarm synchronization using the REST API. For more information about alarm synchronization on the DSC, refer to the [DSC Alarms Guide, Alarm Synchronization](#).

Overview

Alarm synchronization allows all active alarms stored in the Active Trap List on the DSC to be kept in sync with an Element Management System (EMS) or other external client manager, at all times. The synchronization is provided using the REpresentational State Transfer (REST) Application Program Interface (API).

This feature allows the DSC REST API clients to perform the following:

- Query and retrieve specific outstanding alarms, and partial or full active alarm lists on the DSC using HTTP GET.
 - The DSC HTTP response will contain information about the outstanding alarms in the SNMP trap format.
- Request the DSC to clear, acknowledge (ack), unacknowledge (unack), and set comments using HTTP POST.
 - The POST request contains the sequence number as an input to allow the DSC to locate the associated alarm in the list.



Note

To receive an SNMP trap, a host needs to be entered in the Trap Host List. This can be accomplished through the Web UI (refer to [SNMP Configuration](#)) or SNMP if EMS Access is enabled (refer to [Configuring Community Directives](#)).

Enabling Alarm Synchronization on the DSC

To enable the alarm synchronization on the DSC, refer to the .

Once the alarm synchronization is enabled, the DSC stores all active traps in the Web UI under **SNMP > Active Trap List**.

For more information about DSC alarms, refer to the . The Alarms guide describes all events including alarms, warnings, notifications, clearing events, and Intelligent Platform Management Interface (IPMI) hardware sensor alarms recorded by the DSC Platform as well as possible troubleshooting actions.

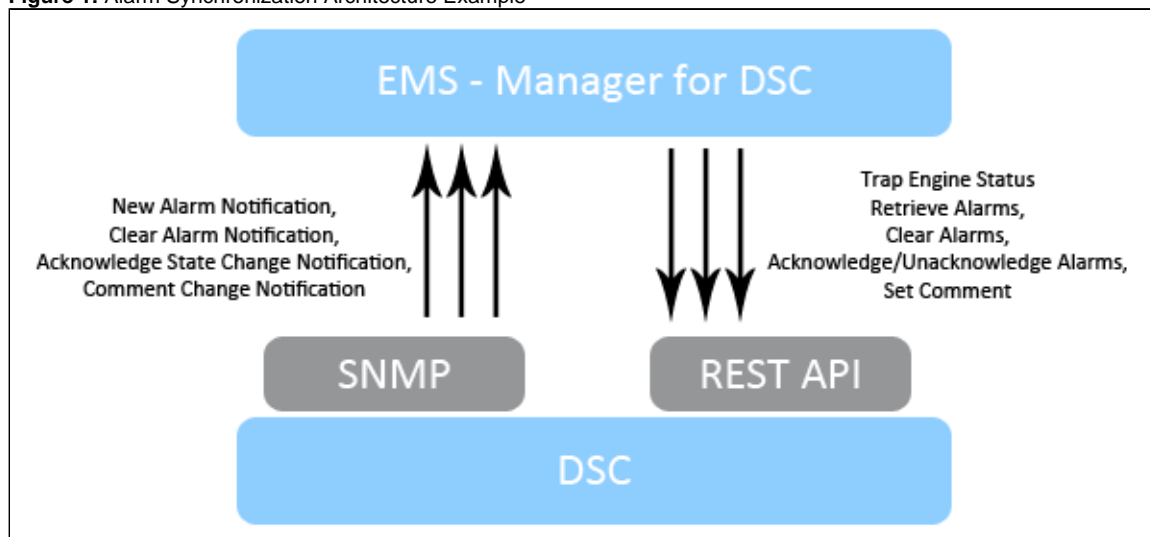
Alarm State Synchronization

The following figure shows an example of the EMS sending requests through the REST API to the DSC, and the DSC sending SNMP messages to the EMS.

Note
The REST API handlers will check that the Active Traps attribute and the Single Trap Stream attribute are set for each incoming HTTP request and will return an error in response to the HTTP request if the DSC is not properly configured for this feature.

Note
The HTTP requests coming from EMS with the goal of alarm synchronization are processed by an addition to DSC REST API - Alarm Sync subsystem.

Figure 1: Alarm Synchronization Architecture Example



Note
If you send an HTTP POST request from the EMS and the network is overloaded or there are connectivity issues, the EMS may not receive the corresponding DSC SNMP Trap response. To check if your request was successful, you can send another request to the DSC using HTTP GET to confirm changes.

Note
If an alarm is manually cleared through the EMS, it will be removed from the Active Trap List and will not be returned in the HTTP GET response, but will still be visible under the cleared bucket list on the DSC Web UI.

Note
If an alarm was manually cleared on the DSC before the Active Traps attribute was enabled, the corresponding clear notification will not be sent to the EMS after the Active Traps attribute is enabled.

If you require a previously cleared alarm to be sent to the EMS, you must manually restore the alarm from the Web UI (refer to [Restoring an Alarm](#)), so the DSC can send a new trap notification to the EMS.

Rest API Commands

The following table lists the REST API commands:

Table 1: REST API Commands

HTTP Command	Resource URL	Category	Comment
GET	/mgmt/rest/trap/getAlarmList	Trap Resync	Returns the list of all the traps available in the DSC.
GET	/mgmt/rest/trap/getAlarmList/{SequenceId}	Trap Resync	Returns the list of the traps available in the DSC using parameter SequenceId, which represents the starting trap of the list.
GET	/mgmt/rest/trap/getAlarmList/{SequenceId}/{count}	Trap Resync	Returns the list of the traps whose size is specified by the count parameter and the starting sequence id specified by the SequenceId parameter.
GET	/mgmt/rest/trap/getCount	Trap Resync	Returns the sum of CRITICAL, MAJOR, MINOR, and WARNING alarms
GET	/mgmt/rest/trap/getLastSequenceId	Trap Resync	Returns the last sequence id of trap sent out.
GET	/mgmt/rest/trap/getTrapDetails/{SequenceId}	Trap Resync	Returns the details of the specified alarm SequenceId.
POST	/api/config/alarms/current/_operations/state/alarmList	Alarm Acknowledge Alarm Unacknowledge	Sets the current acknowledge state and returns a list of successful, failed and not found alarm ids.
POST	/api/config/alarms/current/_operations/comment/alarmList	Alarm Comment	Sets the current comment field and returns a list of successful, failed and not found alarm ids.
POST	/api/config/alarms/clear/_operations/alarmList	Alarm Clear	Removes outstanding alarms and returns a list of successful, failed and not found alarm ids.
GET	/mgmt/rest/trap/getRollOverNumber	Max Sequence ID	DSC sequence number starts at 1. The max sequence number is $2^{31} - 1$. For example <pre><?xml version="1.0" ?> <currentStatus> <rollOverNumber>2147483647</rollOverNumber> </currentStatus></pre>
GET	/mgmt/rest/trap/getTrapEngineStatus	Trap Resync	Trap engine status for EMS to ping. The Timestamp of the trap engine activation has to be monitored for change. For example <pre><?xml version="1.0" ?> <currentStatus> <trapEngineBoot>1496709954</trapEngineBoot> </currentStatus></pre>

