Cisco Prime Central for Hosted Collaboration Solution 9.2.1 Programmer Guide

Cisco Systems, Inc.
www.cisco.com

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco website at www.cisco.com/go/offices.

Text Part Number: OL-29195-01

Cisco Prime Central for Hosted Collaboration Solution 9.2.1 Programmer Guide

CONTENTS

Preface

This guide describes the Prime Central for HCS SNMP and Web Services Northbound Interface APIs, and provides instructions for using them.

Conventions

This document uses the following conventions:

Item	Convention
Commands and keywords	**boldface** font
Displayed session and system information	`screen` font
Information that the user must enter	**`boldface screen`** font
Variables that the user must supply	*`italic screen`* font
Menu items and button names	**boldface** font
Selecting a menu item	**Option > Network Preferences**

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

Timesaver Means *the described action saves time*. You can save time by performing the action described in the paragraph.

Tip Means *the following information will help you solve a problem*.

Product Documentation

Table 1 lists the Prime Central for HCS documentation set.

We sometimes update the documentation after original publication. Therefore, you should also review the documentation on Cisco.com for any updates. You must access the links in Table 1 for the most current Prime Central for HCS 9.2.1 documentation.

Table 1 Related Documentation

Document Title	Available Formats
Prime Central for Cisco Hosted Collaboration Solution 9.2.1 User Guide	On Cisco.com: http://www.cisco.com/en/US/products/ps12491/products_user_guide_list.html
Prime Central for Cisco Hosted Collaboration Solution 9.2.1 Installation Guide	On Cisco.com: http://www.cisco.com/en/US/products/ps12491/prod_installation_guides_list.html
Prime Central for Cisco Hosted Collaboration Solution 9.2.1 Programmer Guide *(this document)*	On Cisco.com http://www.cisco.com/en/US/products/ps12491/prod_technical_reference_list.html
Prime Central for Cisco Hosted Collaboration Solution 9.2.1 Release Notes	On Cisco.com: http://www.cisco.com/en/US/products/ps12491/prod_release_notes_list.html
Open Source Used In Prime Central for Cisco Hosted Collaboration Solution 9.2.1	On Cisco.com: http://www.cisco.com/en/US/products/ps12491/products_licensing_information_listing.html

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

Subscribe to the *What's New in Cisco Product Documentation* as an RSS feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service. Cisco currently supports RSS Version 2.0.

Getting Started

This chapter describes how to use the northbound interface of Prime Central for Cisco Hosted Collaboration Solution (HCS). You can manage services through a variety of associated requests or operations.

This guide also includes feature descriptions, sample Web Service Definition Language (WSDL) extract, typical workflow steps, and other relevant information.

This chapter contains the following sections:

Audience

This guide is intended to be a technical resource for application developers who want to use the northbound interface to retrieve the data from Cisco Hosted Collaboration Solution (HCS) deployments and implementations.

To use this guide, you need an advanced level of understanding of Internet network design, operation, and terminology. You also need to understand the basic concepts of Prime Central for HCS.

You should also understand high level programming language such as Java, and the following:

- XML and XML Schema
- Web Service Definition Language (WSDL)
- Web Services
- Socket programming
- SNMP
- Web Services standards:
  - WS-Notification
  - WS-Enumeration
  - WS-Resources

Feature Summary

Prime Central for HCS offers two types of northbound interfaces:

- SNMP Trap
- Web Services

This document describes both event format and web services interfaces.

Client Requirements

Prime Central for HCS uses Web Services standards. The client must be able to connect to Prime Central for HCS using HTTPS.

For the SNMP trap interface, the client must have SNMP trap listening capability.

Understanding the Web Service Interfaces

Using Prime Central for HCS, you can view the events in a single dashboard that originates from the domain managers.

This chapter contains the following sections:

Interfaces

Prime Central for HCS supports two northbound interfaces (NBI). They are:

- SNMP Notifications—Prime Central for HCS supports SNMP Trap Notifications. SNMP listeners are added, updated, or removed by WebServices API.
- WebServices API—Prime Central for HCS supports the following WebServices APIs:
  - authenticate—authenticate Response, page 3-1
  - getActiveEventCount—getActiveEventCount Response, page 3-2
  - getActiveEvents—getActiveEvents Response, page 3-2
  - getArchivedEventsCount—getArchivedEventsCount Response, page 3-2
  - getArchivedEvents—getArchivedEvents Response, page 3-3
  - getEvent—getEvent Response, page 3-3
  - getOperationalData—getOperationalData Response, page 3-4
  - subscribe—subscribe Response, page 3-4
  - unSubscribe—unSubscribe Response, page 3-4

Data

Incoming events are alarms raised by the underlying domain managers (such as Cisco Unified Operations Manager (CUOM), UCS Manager) that indicates the status of the managed devices. Prime Central for HCS supports various levels of processing for these incoming events and they listed below:

- Normalized-only events—Events that Prime Central for HCS receives and normalizes. These events are not enriched further. It passes on the events directly to a northbound system without additional processing. These events are marked with EventTypeID=default.
- Enriched events—After normalization, some events are enriched with additional information. For example, CUOM events are enriched with the CustomerName, the VM Name in which the UC application is running, and others. Some of the enriched events are used to determine its impact on customer services by overlaying them on the service impact tree.
- Root-cause events—Raw or synthetic events that are the root-cause of the failure.
- Symptomatic events—Raw or synthetic events that are part of event correlation but not the root-cause of the failure.
- Synthetic Events—These are events generated internally to indicate a category of events. For example, all Service Down events on a CUCM node, which indicate that services running on the CUCM node are down, are grouped under the Synthetic event named OM_CUCM_Processed. Synthetic events that participate in the correlation tree are used for root cause analysis.
- Service-impact events—Service-impact events describe the state of services; this is an event generated to notify the state of the top node in the service impact tree.

For more information on the enrichment level, see table Table 2-1. For more information on individual field names in an NBI event notification, see table Table 2-2.

When symptom and root-cause events arrive, it is possible that a symptom event is marked as a root-cause event until such time as the root-cause event arrives. After the real root-cause event arrives, the CauseType of the symptom event changes to symptom.

If you have set subscription filters based on CauseType, you will receive the symptom event as a root-cause event. However, this will later be updated with the CauseType Symptom.

Table 2-1 Level of Event Enrichment Based on Various Parameters

Field Name	Normalized-Only Events	Enriched-Only Events	Root-Cause Events	Symptomatic Events	Service Impact Events
EventIdentifier	Yes	Yes	Yes	Yes	Yes
EventName	Yes (only for CUOM, UCSM, Infrastructure Monitoring)	Yes (only for CUOM, UCSM, Infrastructure Monitoring)	Yes (same as EventType ID)	Yes (same as EventType ID)	—
Summary	Yes	Yes	Yes	Yes	Yes

Table 2-1 Level of Event Enrichment Based on Various Parameters (continued)

Field Name	Normalized-Only Events	Enriched-Only Events	Root-Cause Events	Symptomatic Events	Service Impact Events
ComponentId	Yes (only for CUOM, UCSM, Infrastructure Monitoring)	Yes (only for CUOM, UCSM, Infrastructure Monitoring)	Yes (only vCenter and UCSM Events)	Yes (only vCenter and UCSM Events)	—
DeviceId	Yes	Yes	Yes	Yes	—
DomainManagerID	Yes	Yes	Yes	Yes	—
Customer	—	Yes (Only OM & Infrastructure Monitoring VM events)	Yes (Only OM & Infrastructure Monitoring VM events)	Yes (Only OM & Infrastructure Monitoring VM events)	Yes
CustomerExtName	—	Yes (Only OM & Infrastructure MonitoringVM events)	Yes (Only OM & Infrastructure MonitoringVM events)	Yes (Only OM & Infrastructure Monitoring VM events)	Yes
CauseType	Yes (set to Unknown)	Yes (set to Unknown)	Yes (set to Rootcause)	Yes (set to Symptom)	—
ParentEventID	—	—	Yes	Yes	—
Severity	Yes	Yes	Yes	Yes	Yes
OriginalSeverity	Yes	Yes	—	—	—
EventTypeId	—	Yes	Yes	Yes	—
ProblemeventID	Yes	Yes	Yes	Yes	Yes
ServiceName (for service events only)	—	—	—	—	Yes
ServiceImpactType (for service events only)	—	—	—	—	Yes
OperationalDataPointer	Yes (only for CUOM, UCSM, Infrastructure Monitoring)	Yes (only for CUOM, UCSM, Infrastructure Monitoring)	—	—	—
Dashboard URL	Yes	Yes	Yes	Yes	Yes

Table 2-1 Level of Event Enrichment Based on Various Parameters (continued)

Field Name	Normalized-Only Events	Enriched-Only Events	Root-Cause Events	Symptomatic Events	Service Impact Events
Count	Yes	Yes	—	—	—
Last Occurrence	Yes	Yes	Yes	Yes	Yes
Event Status	Yes	Yes	Yes	Yes	

This section explains the formats of the incoming SNMP v2C Trap and description of the associated variable bindings. This table also indicates the fields or variable bindings that can be used to filter the incoming data.

Table 2-2 NBI Event Format

Field Name	Filterable	Field Description	Field Type	Field Value	Varbind OID
EventName	Yes	Name of the event	OctetString	• Synthetic events: HCM defined EventName (similar to EventTypeID) • UCSM events: text representation of cucsFaultCode enum • CUOM events: EventName	1.3.6.1.4.1.1279.1
Summary	No	Brief description of the event	OctetString	• Synthetic events: HCS defined event summary • CUOM events: AlarmDescription • UCSM events: cencucsFaultDescription	1.3.6.1.4.1.1279.2

Table 2-2 NBI Event Format (continued)

Field Name	Filterable	Field Description	Field Type	Field Value	Varbind OID
ComponentId	Yes	Name/identifier of the component within device that event is raised for (For example, UCS blade)	OctetString	• Synthetic events: take below fields from original domain manger event • CUOM events:cenAlarmManagedObjectClass • UCSM events: cucsFaultAffectedObjectDn	1.3.6.1.4.1.1279.3
DeviceId	Yes	Hostname or IP of device that originated event (For example, CUCM, Router, etc.)	OctetString	—	1.3.6.1.4.1.1279.4
Domain ManagerID	Yes	IP address of domain manager that sent event to Prime Central for HCS (For example, CUOM, UCSM, etc.)	OctetString	—	1.3.6.1.4.1.1279.5
Customer	Yes	Customer Name	OctetString	—	1.3.6.1.4.1.1279.6
Customer ExtName	Yes	Customer Name used in MSP external CMDBs (stored in HCS CDM)	OctetString	—	1.3.6.1.4.1.1279.7

Table 2-2 NBI Event Format (continued)

Field Name	Filterable	Field Description	Field Type	Field Value	Varbind OID
CauseType	Yes	Flag indicating whether event is the root cause or symptomatic event • 0—Unknown • 1—Root cause • 2—Symptom number True for events identified as root-cause and marked only after root cause is finalized, that is after the RCA timer expires. False for others.	Gauge32	—	1.3.6.1.4.1.1279.8
ParentEvent ID	No	EventID pointing to parent event in correlation tree – can be used by NB system to reconstruct Prime Central for HCS event correlation tree for this event	OctetString	—	1.3.6.1.4.1.1279.9
Severity	Yes	Event severity assigned by Prime Central for HCS: • 0—Clear • 1—Indeterminate • 2—Warning • 3—Minor • 4—Major • 5—Critical	Gauge32	For cleared events severity = 0; For active events, see Prime Central for HCS and Domain Manager Severity Mapping	1.3.6.1.4.1.1279.10
Original Severity	No	Original event severity assigned by domain manager.	OctetString	—	1.3.6.1.4.1.1279.11

Table 2-2 NBI Event Format (continued)

Field Name	Filterable	Field Description	Field Type	Field Value	Varbind OID
EventTypeId	No	ID used to group domain manager events with the similar impact on component state for common Prime Central for HCS processing. For example, UCS_Blade_Availability includes all events causing blade failure.	OctetString	See EventTypeID Mapping.	1.3.6.1.4.1.1279.12
Problemevent ID	No	Used for clearing event. Refers to previous problem event which it clears.	OctetString	EventID of the event that is being cleared	1.3.6.1.4.1.1279.13
ServiceName (for service events only)	Yes	Name of Prime Central for HCS service that event is raised for (For example, Customer Voice).	OctetString	Name of top node of service impact tree (for example, Customer Voice Service).	1.3.6.1.4.1.1279.14
ServiceImpactType (for service events only)	Yes	Service state; state of the service with ServiceName.	OctetString	State of the top node of instance of service impact tree (for example, state of Customer Voice Service for customer X); Values can be: UP, MARGINAL or DOWN	1.3.6.1.4.1.1279.15
OperationalDataPointer (for RC events only)	No	Pointer to the knowledge base/reference guides with next steps information	OctetString	URL to specific event as specified in domain manager event reference guide	1.3.6.1.4.1.1279.16
Count	No	Number of times this event has occured.	Gauge32	—	1.3.6.1.4.1.1279.17

Table 2-2 NBI Event Format (continued)

Field Name	Filterable	Field Description	Field Type	Field Value	Varbind OID
Type	Yes	• 1—More Severe • 2—Less Severe • 3—Information	Gauge32	—	1.3.6.1.4.1.1279.18
Last Occurrence	No	Timestamp indicating when the event last occurred. The value is represented in seconds since epoch.	Gauge32	—	1.3.6.1.4.1.1279.19
PrimeGuiUrl	No	Event URL.	OctetString	—	1.3.6.1.4.1.1279.20
EventIdentifier	Yes	Unique ID for each event.	OctetString	—	1.3.6.1.4.1.1279.21
ContainerId	Yes	Synthetic Event Identifier for Raw events.	OctetString	—	1.3.6.1.4.1.1279.22
FirstOccurrence	No	Timestamp indicating when event first occurred. The value is represented in seconds since epoch.	Gauge32-	—	1.3.6.1.4.1.1279.23

The following table explains the mapping of severity between Prime Central for HCS and domain managers.

Table 2-3 Prime Central for HCS and Domain Manager Severity Mapping

Prime Central for HCS Severity	CUOM Severity	UCSM Severity	DCSM-SAN Severity	DCNM-LAN Severity	Infrastructure Monitoring Severity
0—Clear, Green	N/A	0—Clear	—	—	—
1—Indeterminate, Purple	N/A	1—Info	Debugging	Debugging	Informational
2—Warning, Blue	Informational	3—Warning	Info, Notification	Info, Notification	Harmless
3—Minor, Yellow	Warning	4—Minor	Warning	Warning	Warning
4—Major, Orange	N/A	5—Major	Alert, Emergencies	Alert, Emergencies	—
5—Critical, Red	Critical	6—Critical	Error, Critical	Error, Critical	Critical

Filtering

Using the filtering arguments available in the Prime Central for HCS Northbound API you can retrieve event data based on your requirements. For information on examples of usage of whereclause, see Examples of Usage of whereclause, page 3-5.

To see if the data is filterable, see table Table 2-2. You can set filters based on the parameters outlined in the table.

Rules for Writing Filters

There are three categories of filters associated with the NBI. They are:

- Gateway filter—Used in the subscribe API of the NBI. Gateway filter defines which of the events should be forwarded to the destination.
- Active event retrieval filter—Used in the getActiveEvents and getActiveEventCount API of the NBI. This filter defines which of the events should be retrieved from the Active event database.
- Archive event retrieval filter—Used in the getArchivedEvents and getActivedEventCount API of the NBI. This filter defines which of the events should be retrieved from the Archived event database.

The following restrictions apply:

- Filters follow the format of a SQL where clause. For more information, see chapter WSDL Specifications.
- You can specify only those columns that are marked as filterable and they are listed in Table 2-2.

Note The Fieldnames are case-sensitive. Specify exactly as mentioned in the Table 2-2.

- Review the Field type to understand if a column is of type Number or String. The filter expression syntax is different for these two types of variables.
- It may also be necessary to review the expected set of values for the columns that are of type Number. For example, for active events, you must specify EventStatus=1. For symptomatic events, you must specify CauseType=2.
- Do not use Ampersand (&) to filter data.

The following rules apply when specifying where clause conditions:

- Where clause can include logical operators such as AND and OR.
- Where clause can include standard comparison operators such as <, >, <=, >=, =, !=, <>
- Oracle-specific conditional clauses *is null* or *not null* are not supported.
- Use column != ' ' to get expected output.
- IN clause is supported.
- LIKE clause is supported with an exception that Percentage (%) and Underscore (_) are treated as literals.
- String (VARCHAR) values within WHERE condition must be specified within single quotes (for example, DeviceId='CUCM-1').
- Numbers within WHERE condition must be specified without any quotes (for example, Severity=5)

WSDL Specifications

You can develop web-service clients using the WSDL API methods explained below. The WSDL is available at https://<*prime-central-ip-address*>:9090/EventManagement?wsdl.

This chapter contains the following sections.

WSDLs in Prime Central for HCS

This section explains the WSDL API methods that you can leverage. The **whereclause** is explained in Filtering, page 2-9. For whereclause examples, see Examples of Usage of whereclause, page 3-5.

The following operations are applicable for Prime Central for HCS:

authenticate—authenticate Response

This WSDL API method authenticates users against Prime Central for HCS. In response, it provides a session token to be used for subsequent requests.

The following table describes the arguments used with the WSDL API method:

Argument	Description
username	Username for Prime Central for HCS
password	Password for the above user.

getActiveEventCount—getActiveEventCount Response

Returns the count of rows that contain active events. The response is a number that represents the number of active events.

The following table describes the arguments used with the WSDL API method:

Argument	Description
whereclause	Filter in SQL format.
sessionToken	Session ID obtained using authenticate.

The following table provides the description of return values:

Return Value	Description
NumberOfActiveEvents	The number of events currently active (uncleared) in the system.

getActiveEvents—getActiveEvents Response

This operation returns the available active events. The response to this request is a list of all active events. The following table describes the arguments used in the request:

Argument	Description
whereClause	Filter in SQL format.
sessionToken	Session ID obtained using authenticate.

getArchivedEventsCount—getArchivedEventsCount Response

This request gets the number of archived events from archive database. The response is a number that represents the number of archived events.

The following table describes the arguments used in the WSDL API method:

:

Argument	Description
whereclause	Filter in SQL format.
sessionToken	Session ID obtained using authenticate.

The following table provides the description of return values:

Return Value	Description
ArchivedEventsCount	Count of the number of archived events.

getArchivedEvents—getArchivedEvents Response

The request can be used to retrieve archived events from the archive database. The response is a list of all archived events in the archive database.

The following table describes the arguments used with the WSDL API method:

:

Argument	Description
whereclause	Filter in SQL format.
sessionToken	Session ID obtained using authenticate.

The following table provides the description of return values:

Return Value	Description
ArchivedEventsList	List of archived events.

getEvent—getEvent Response

This request retrieves the active events present in the database. The response provides active or archived events present in the database.

The following table describes the arguments used with the WSDL API method:

Argument	Description
EventId	EventIdentifier of the event.
sessionToken	Session ID obtained using authenticate.

The following table provides the description of return values:

Return Value	Description
Event	The event that matches the specified EventId.

getOperationalData—getOperationalData Response

This request serves the next steps URL or HTML fragment. The response is a URL to a page that contains next steps or a HTML fragment that contains next steps.

The following table describes the arguments used with the WSDL API method:

Argument	Description
eventName	EventName of an event. For example, PerformancePollingStopped,SysLogIdentifier
sessionToken	Session ID obtained using authenticate.

subscribe—subscribe Response

This request lets you subscribe to receive SNMP traps for events. The response returns SubscriptionID. This SubscriptionID must be used to unsubscribe. For example, the SubscriptionID can be Customer_99.

The following table describes the arguments used in the WSDL:

Argument	Description
snmpgatewayAddr	IP address of the device to which the SNMP traps have to be sent.
snmpGatewayPort	Port number to which the SNMP traps have to be sent.
whereClause	Filter in SQL format.
sessionToken	Session ID obtained by using authenticate.

unSubscribe—unSubscribe Response

Performs the request to unsubscribe a previously subscribed request.

The following table describes the arguments used with the WSDL API method:

Argument	Description
SubscriptionID	SubscriptionID obtained using a subscribe request.
sessionToken	Session ID obtained using authenticate.

Exception Handling

Any error encountered by the NBI while processing the request is returned to the NBI client through a generic EventManagementException object. This object includes an error message describing the specific problem that has occurred. Few example are listed below:

Error Message `The requested operation could not be completed due to a system error:Not a valid session, Please re-authenticate.`

Explanation You specified an invalid sessionToken in the API invocation, and an EventManagementException object is returned.

Error Message `Error Message: The requested operation could not be completed due to a system error:Incorrect result size: expected 1, actual 0.`

Explanation You attempted to unsubscribe with an incorrect SubscriptionID.

Recommended Action Specify the correct SubscriptionID.

Examples of Usage of whereclause

This section presents some examples of the usage of whereclause. For more information on the usage, see Filtering, page 2-9. You can filter based on various parameters, and some examples are outlined below:

Specifying filter for GetEvents

- getActiveEvents using filter whereClause = DeviceId (send only events originated by specific CUCM)

```
<whereClause>Node='CUCM-1'</whereClause>
```

- getActiveEvents using filter whereClause = Severity (send only critical Prime Central for HCS events)

```
<whereClause>Severity=5</whereClause>
```

- getActiveEvents using filter whereClause = Event status (cleared) (send only cleared events)

```
<whereClause>type=2</whereClause>
```

- getActiveEvents using filter whereClause = Event status (active) (example, send only active events)

```
<whereClause>type=1</whereClause>
```

Specifying filter for Subscription

- subscribe SNMP traps with snmpgatewayAddr and snmpGatewayPort for root-cause events

```
<whereClause>CauseType=1</whereClause>
```

- subscribe SNMP traps with snmpgatewayAddr and snmpGatewayPort for Symptomatic events

```
<whereClause>CauseType=2</whereClause>
```

Example of Java language Binding File

The following is an example of a Java language binding generated for the WSDLs above. You have the option of using a language of your choice to create VarBinds. You can create the Java source using the SOAP engines CXF or Muse/Axis2. For more information on filtering events, see Filtering, page 2-9.

```
package com.cisco.hcm.fronthandler;

import java.util.List;

import javax.jws.WebParam;
import javax.jws.WebService;
import javax.xml.bind.annotation.XmlElement;

import com.cisco.hcm.events.Event;
import com.cisco.hcm.utils.EventManagementException;

@WebService(name = "EventManagement", targetNamespace =
"http://fronthandler.hcm.cisco.com/")
public interface EventManagement {

    /**
     *
     * @param whereClause
     * @param sessionToken
     * @return
     * @throws EventManagementException
     * Returns the active events in Omnibus objectserver
     */
    public List<Event>
getActiveEvents(@XmlElement(required=true)@WebParam(name="whereClause")String whereClause
,@XmlElement(required=true)@WebParam(name="sessionToken")String sessionToken) throws
EventManagementException;

    /**
     *
     * @param whereClause
     * @param sessionToken
     * @return
     * @throws EventManagementException
     * Returns the Archived events
     */
    public List<Event>
getArchivedEvents(@XmlElement(required=true)@WebParam(name="whereClause")String
whereClause , @XmlElement(required=true)@WebParam(name="sessionToken")String sessionToken)
throws EventManagementException;

    /**
     *
     * @param eventId
     * @param sessionToken
     * @return
     * @throws EventManagementException
     */
    public Event getEvent(@XmlElement(required=true)@WebParam(name="eventId")String
eventId , @XmlElement(required=true)@WebParam(name="sessionToken")String sessionToken)
throws EventManagementException;

    /**
     *
     * @param whereClause
```

```
     * @param sessionToken
     * @return
     * @throws EventManagementException
     * Return Count of the rows based on filter
     */
    public int
getActiveEventCount(@XmlElement(required=true)@WebParam(name="whereClause")String
whereClause , @XmlElement(required=true)@WebParam(name="sessionToken")String sessionToken)
throws EventManagementException;

    /**
     *
     * @param whereClause
     * @param sessionToken
     * @return
     * @throws EventManagementException
     * Return Count of the rows based on filter
     */
    public int
getArchivedEventCount(@XmlElement(required=true)@WebParam(name="whereClause")String
whereClause , @XmlElement(required=true)@WebParam(name="sessionToken")String sessionToken)
throws EventManagementException;

    /**
     *
     * @param snmpgatewayAddr
     * @param snmpGatewayPort
     * @param whereClause
     * @param sessionToken
     * @return
     * @throws EventManagementException
     * Serve the Subscribe request
     */
    public String
subscribe(@XmlElement(required=true)@WebParam(name="snmpgatewayAddr")String
snmpgatewayAddr, @XmlElement(required=true)@WebParam(name="snmpGatewayPort")String
snmpGatewayPort,
            @XmlElement(required=true)@WebParam(name="whereClause")String whereClause ,
@XmlElement(required=true)@WebParam(name="sessionToken")String sessionToken) throws
EventManagementException;

    /**
     *
     * @param customerToken
     * @param sessionToken
     * @return
     * @throws EventManagementException
     * Serve the Un-Subscribe request
     */
    public String
unSubscribe(@XmlElement(required=true)@WebParam(name="customerToken")String customerToken
, @XmlElement(required=true)@WebParam(name="sessionToken")String sessionToken) throws
EventManagementException;

    /**
     *
     * @param eventCode
     * @param sessionToken
     * @return
     * @throws EventManagementException
     * Serve the next steps request
     */
```

```
    public String
getOperationalData(@XmlElement(required=true)@WebParam(name="eventName")String eventCode ,
@XmlElement(required=true)@WebParam(name="sessionToken")String sessionToken) throws
EventManagementException;

    /**
     *
     * @param username
     * @param password
     * @return
     * @throws EventManagementException
     * Provides sessionToken to be used for other requests
     */
    public String authenticate(@XmlElement(required=true)@WebParam(name="username")String
username, @XmlElement(required=true)@WebParam(name="password")String password) throws
EventManagementException;

}
```

Event Management NBI

The enhanced Event Management Northbound Interface (NBI) supports multiple SNMP gateway subscriptions. The maximum subscriptions allowed at one time is five. The gateway subscriptions support the ability to filter the incoming traps. A Python client script supports subscribe and unsubscribe calls into the Event Management NBI. In addition, MIB definition for SNMP notifications generated by the NBI is also supported.

This chapter contains the following sections:

Executing Client Script

Once you run the patch script, the client scripts are installed on Prime Central VM.

The scripts are located at the following location in the Prime Central VM: **<*install_dir*>/prime-hcs/scripts/pc4hcs_nbi_client.py.**

This section explains the various arguments and scenarios involved in subscribing and unsubscribing to traps. This section contains the following topics:

Subscribing

To subscribe, the following arguments are required:

- subscribe—Specifies that this is a subscribe command
- prime-ip—IP or hostname of the Prime Central VM
- prime-admin-password—Prime password used at the time of installation of Prime VM
- snmp-gateway-ip—IP or hostname to which traps should be sent
- snmp-gateway-port—Port to which traps should be sent

- where-clause—whereclause to set the filter on; this is optional

Step 1 Log in as root user.

Step 2 Run the following command:

```
./pc4hcs_nbi_client.py [--subscribe | --unsubscribe]
                       --prime-ip
                       --prime-password
                       --where-clause
                       --snmp-gateway-port
                       --snmp-gateway-ip
```

The request returns the **customer-*token***.

Here is an example of subscribe request:

```
[root@hcslab-mike pc4hcs_nbi_client]# ./pc4hcs_nbi_client.py --subscribe --prime-ip
10.10.10.10 --prime-password Admin123@ --snmp-gateway-port 162 --snmp-gateway-ip
20.20.20.20 --where-clause "Customer = 'ABC'"
```

The following is an example of the returned subscription token:

```
Customer_9
```

Note Retain the subscription token value to unsubscribe at a later time.

Using Where Clause for Subscribe Request

Using the where clause argument is optional; however, if you do not use the where clause, all the traps, without any filtering, will be forwarded through the gateway. If you do specify the where clause, only traps that match the where clause criteria will be sent for the subscription. The argument snmp-gateway-ip specifies the destination IP to which traps will be sent.

Unsubscribing

To unsubscribe, the following arguments are required:

- unsubscribe—Specifies that this is an unsubscribe command
- prime-ip—IP or hostname of the Prime Central server
- prime-admin-password—Prime admin password used at the time of installation of Prime Central VM
- customer-token—The subscription token that was returned when you subscribed to receiving traps. If you do not have the subscription-token, follow the steps outlined in the section Recovering Subscription Tokens, page 4-3

Step 1 Log in as root user.

Step 2 Run the following command:

```
./pc4hcs_nbi_client.py [--subscribe | --unsubscribe]
                       --prime-ip
                       --prime-password
                        --customer-token
```

The following message is returned:

```
Un-Subscribe request successful.
```

Here is an example of unsubscribe request:

```
[root@hcslab-mike pc4hcs_nbi_client]# ./pc4hcs_nbi_client.py --unsubscribe --prime-ip
172.16.10.126 --prime-password Admin123@ --customer-token Customer_9
```

The following message is returned:

```
Un-Subscribe request successful.
```

Recovering Subscription Tokens

When you subscribe to receive traps by executing a subscribe request, a subscription token is returned. This token must be retained to unsubscribe at a later time. If you do not have the subscription token, follow these steps to recover the token:

1. To recover the token, you need to be logged into a VM that has Prime Central installed.
2. You must log into Oracle as either oracle user or switch user (su), and then query the primedb database.
3. Specify the `--prime-admin-password` you specified at the time of installation.

Step 1 Log on to the Prime Central VM.

Step 2 Run the following command:

```
su - oracle -c '$ORACLE_HOME/bin/sqlplus primedba/\"<prime-admin-password>\"@primedb'
```

The query command returns the ID, destination, status, and customer token. Match the customer token with the destination you wish to unsubscribe.

The following values indicate the status of the gateway:

- 1—The gateway is up
- 0—The gateway is down

Here is an example of recovery command:

```
[root@hcslab-mike pc4hcs_nbi_client]# su - oracle -c '$ORACLE_HOME/bin/sqlplus
primedba/\"Admin123@\"@primedb'
```

The following values are returned:

```
SQL> select * from SNMPGATEWAYS;

                ID DESTINATION                    STATUS CUSTOMERTOKEN
---------- ------------------------------ ----------
--------------------------------------------
```

```
1 172.16.10.217:163                          1 Customer_57

2 0

3 0

4 0

5 0
```

Recovering Active Subscription and Nonactive Gateway

Follow the steps explained below to recover active subscription and nonactive gateway:

Step 1 From Prime Central VM, log on to the Oracle database.

Step 2 Enter this command for updating the DB:

update SNMPGATEWAYS set DESTINATION=null, STATUS=0, CUSTOMERTOKEN=null where ID=*active-subscription-ID*;

Step 3 Login in as a root user on a Event Collector VM and remove the following files:

If the destination ID is 1, use

```
rm -rf $OMNIHOME/etc/NCO_GATE.trapdest
rm -rf $OMNIHOME/var/NCO_GATE.pid
```

If the destination ID is a value other than 1, use

```
rm -rf $OMNIHOME/etc/NCO_GATE_trap-destination-ID.trapdest
rm -rf $OMNIHOME/var/NCO_GATE_trap-destination-ID.pid
```

Note Before deleting the *NCO_GATE_trap-destination-ID.pid*, use the **cat** command to determine the PID of the process you want to kill and verify the details returned from the command in Step 4.

Step 4 Enter the kill command if the gateway is still up:

```
ps -ef | grep snmp
kill -9 <pid of gateway to be brought down>
```

Tip Identify the PID from the gateway line, when you enter the command **ps -ef | grep snmp**:

An example with PID 30443 is listed below:

```
netcool 30445 30443 0 12:49 ? 00:00:00
/opt/IBM/tivoli/netcool/omnibus/bin/linux2x86/nco_g_snmp -name NCO_GATE -snmpgateway
sadev-rod:166
```

Do not terminate this process; it does not stop the gateway:

```
netcool 30443 25808 0 12:49 ? 00:00:00 /bin/bash
/opt/IBM/tivoli/netcool/omnibus/bin/hcs_snmp_gateway_start 1
```

Recovering From Inconsistent Oracle DB State

Sometimes, when you enter a subscription or unsubscription request, your machine hangs and times out. In such a scenario, determine if the Oracle database is in an inconsistent state:

Step 1 Open the log **/opt/prime/esb/data/log/servicemix.log**. Look for the following lines:

```
OALL8 is in an inconsistent state; Nested exception is java.sql.SQLException: OALL8 is in
an inconsistent state.
```

Step 2 If the Oracle database is in an inconsistent state reboot it using the following command:

```
su - primeusr -c 'emdbctl stop; emdbctl start'
```

MIB Definition for NBI SNMP Notification

After you run the patch script, the MIB file is installed on Prime Central VM. The MIB defines an SNMP v2 notification that is aligned with the SNMP v2 notifications generated by Prime Central for HCS.

The MIB file is located at the following location in the Prime Central VM: <*prime-install-directory*>**/prime-hcs/sdk/CISCO-HCS-SA-NBI-MIB.my.**

Northbound clients that intend to process the SNMP notifications from Prime Central for HCS can use the MIB definition to implement this feature.

For more information, see the documentation that ships with the product used to process the received SNMP notifications.

EventTypeID Mapping

This appendix lists the EventTypeID associated with a particular event categorized based on domain manager. This helps you filter the events using EventTypeID. Prime Central for HCS sets the EventTypeID to Default for all normalized-only events. In case of normalized-only events, Prime Central for HCS displays the EventName and the ComponentID only if the event originates from CUOM, UCSM, and Infrastructure Monitoring.

The events originate from the following domain managers:

CUOM Events

This section outlines the EventTypeIDs for events originating from the domain manager CUOM. It covers the following topics:

Table A-1 CUOM for CUCM Events

Event	Domain Manager Severity	EventTypeID
ServiceDown	Critical	OM_CUCM_Processes
DeviceRestarted	Informational	OM_CUCM_NodeRestart
Unresponsive	Critical	OM_CUCM_OM_Connectivity
HTTPInaccessible	Critical	OM_CUCM_Connectivity
PerformancePollingStopped	Critical	OM_CUCM_Connectivity

Table A-1 CUOM for CUCM Events (continued)

Event	Domain Manager Severity	EventTypeID
SDLLinkOutOfService	Warning	OM_CUCM_Redundancy
SystemVersionMismatched	Informational	OM_CUCM_Redundancy
UnknownPublisher	Critical	OM_CUCM_Redundancy
CTILinkDown	Critical	OM_CUCM_Redundancy
Number Of Registered Phones Dropped	Warning	OM_CUCM_Registration
PhoneUnregistered	Warning	OM_CUCM_Registration
PhonesUnregisteredThreshold Based	Warning	OM_CUCM_Registration
PhoneUnregThresholdExceede d	Critical	OM_CUCM_Registration
Number Of Registered Gateways Increased	Informational	OM_CUCM_Registration
Number Of Registered Gateways Decreased	Informational	OM_CUCM_Registration
Number Of Registered MediaDevices Decreased	Informational	OM_CUCM_Registration
Number Of Registered MediaDevices Increased	Informational	OM_CUCM_Registration
EndPointLostContact	Critical	OM_CUCM_Endpt_ Connectivity
ServiceQualityThresholdCross ed	Critical	OM_CUCM_VoiceQuality_UC M
ServiceQualityThresholdCross ed	Critical	OM_CUCM_VoiceQuality_Devi cePool
ServiceQualityIssue	Critical	OM_CUCM_VoiceQuality_Call Leg
Authentication_Failed	Warning	OM_CUCM_Security
Cisco DRF Failure	Critical	OM_CUCM_BackupRestore
IDS Replication Failure	Critical	OM_CUCM_BackupRestore
DBReplicationFailure	Critical	OM_CUCM_BackupRestore
CDR Agent Send File Failed	Critical	OM_CUCM_CDR
CDR File Delivery Failed	Warning	OM_CUCM_CDR
CDR High Water Mark Exceeded	Warning	OM_CUCM_CDR
CDR Maximum Disk Space Exceeded	Critical	OM_CUCM_CDR
Code-Red	Critical	OM_CUCM_App_Resources
Code-Yellow	Critical	OM_CUCM_App_Resources

Table A-1 CUOM for CUCM Events (continued)

Event	Domain Manager Severity	EventTypeID
CPUPegging	Critical	OM_CUCM_App_Resources
InsufficientFreeMemory	Critical	OM_CUCM_App_Resources
InsufficientFreeHardDisk	Critical	OM_CUCM_App_Resources
InsufficientFreeVirtualMemory	Critical	OM_CUCM_App_Resources
Thread Counter Update Stopped	Informational	OM_CUCM_App_Resources
SIP Trunk Out Of Service	Critical	OM_CUCM_SIP_Trunks
SIP Trunk Partially In Service	Warning	OM_CUCM_SIP_Trunks
Media List Exhausted	Critical	OM_CUCM_CC_Resources
Route List Exhausted	Critical	OM_CUCM_CC_Resources
HighAnalogPortUtilization	Critical	OM_CUCM_CC_Resources
HighDigitalPortUtilization	Critical	OM_CUCM_CC_Resources
HighResourceUtilization	Critical	OM_CUCM_CC_Resources
LocationBWOutOfResources	Critical	OM_CUCM_BW_Resources

Table A-2 CUOM for CUCxN Events

Event	Domain Manager Severity	EventTypeID
ServiceDown	Critical	OM_CUCxn_Processes
Unresponsive	Critical	OM_CUCxn_OM_Connectivity
PerformancePollingStopped	Critical	OM_CUCxn_Connectivity
AutoFailoverFailed	Warning	OM_CUCxn_Redundancy
NoConnectionToPeer	Warning	OM_CUCxn_Redundancy
ReplicationStopped	Warning	OM_CUCxn_Redundancy
CPUUtilizationExceeded	Warning	OM_CUCxn_App_Resources
InsufficientFreeHardDisk	Critical	OM_CUCxn_App_Resources
InsufficientFreeVirtualMemory	Critical	OM_CUCxn_App_Resources
LicenseExpired	Critical	OM_Voicemail_Resources
LowAvailableInboxLicenses	Critical	OM_Voicemail_Resources
LowAvailableSubscriberLicenses	Critical	OM_Voicemail_Resources

Table A-3 CUOM for CUP Events

Event	Domain Manager Severity	EventTypeID
ServiceDown	Critical	OM_CUP_Processes
Unresponsive	Critical	OM_CUP_OM_Connectivity
PerformancePollingStopped	Critical	OM_CUP_Connectivity

Table A-4 CUOM for CVP Events

Event	Domain Manager Severity	EventTypeID
ServiceDown	Critical	OM_CVP_Processes
InsufficientFreeHardDisk	Critical	OM_CVP_App_Resources
HighUtilization-(Processor)	Critical	OM_CVP_App_Resources
Unresponsive	Critical	OM_CVP_Servicability
TemperatureHigh	Critical	N/A for HCS
InsufficientFreeVirtualMemory	Critical	OM_CVP_App_Resources
InsufficientFreeMemory	Critical	OM_CVP_App_Resources

Table A-5 CUOM for UCCE Events

Event	Domain Manager Severity	EventTypeID
HighUtilization	Critical	OM_UCCE_App_Resources
InsufficientFreeMemory	Critical	OM_UCCE_App_Resources
pimDown	Critical	OM_UCCE_Processes
ServiceDown	Critical	OM_UCCE_Processes

Table A-6 CUOM for IOS Routers (ASR1000, ISR, ISR-G2) Events

Event	Domain Manager Severity	EventTypeID
Unresponsive	Critical	OM_ROUTER_Availability
Flapping	Critical	OM_ROUTER_CardAvailability OM_ROUTER_InterfaceAvailability
OperationallyDown	Critical	OM_ROUTER_CardAvailability OM_ROUTER_InterfaceAvailability
Duplicate IP Address	Critical	OM_ROUTER_InterfaceAvailability
TemperatureHigh	Critical	OM_ROUTER_Environmental

Table A-6 CUOM for IOS Routers (ASR1000, ISR, ISR-G2) Events (continued)

Event	Domain Manager Severity	EventTypeID
StateNotNormal(temp, voltage, power, fan)	Critical	OM_ROUTER_Environmental
HighUtilization	Critical	OM_ROUTER_Resource
InsufficientFreeMemory	Critical	OM_ROUTER_Resource
RepeatedRestarts	Critical	OM_ROUTER_Availability

Table A-7 CUOM for CUBE-SP Events

Event	Domain Manager Severity	EventTypeID
Adjacency Detached	Critical	OM_CUBE_SP_AdjacencyStatus
MOSCQEReachedCriticalThreshold	Critical	OM_CUBE_SP_QOS_Critical
MOSCQEReachedMajorThreshold	Warning	OM_CUBE_SP_QOS_Major
MOSCQEReachedMinorThreshold	Informational	OM_CUBE_SP_QOS_Minor
ServiceCardOffline	Critical	OM_CUBE_SP_Status
ServiceCardStandBy	Critical	OM_CUBE_SP_Status
SourceAlert	Critical	OM_CUBE_SP_Security
DynamicBlackList	Warning	OM_CUBE_SP_Security
CPUCongestion	Critical	OM_CUBE_SP_Resource
MemoryCongestion	Critical	OM_CUBE_SP_Resource
SLAViolation	Warning	OM_CUBE_SP_SLA

UCSM Events

This section outlines the EventTypeIDs for events originating from the domain manager UCSM. It covers the following sections:

- "UCSM Events—Processed for Root Cause"
- "UCSM Events—Enriched and Normalized"

Table A-8 UCSM Events—Processed for Root Cause

Event	Domain Manager Severity	EventTypeID
fltLsServerRemoved	Critical	UCS_Blade_Avlblty
fltComputePhysicalAssignedMissing	Critical	UCS_Blade_Avlblty

Table A-8 UCSM Events—Processed for Root Cause (continued)

Event	Domain Manager Severity	EventTypeID
fltComputeBoardPowerError	Major	UCS_Blade_Avlblty
fltComputePhysicalPowerProblem	Major	UCS_Blade_Avlblty
fltComputePhysicalThermalProblem	Major	UCS_Blade_Avlblty
fltComputePhysicalBiosPostTimeout	Critical	UCS_Blade_Avlblty
fltComputePhysicalDiscoveryFailed	Major	UCS_Blade_Avlblty
fltComputePhysicalAssociationFailed	Major	UCS_Blade_Avlblty
fltComputePhysicalInoperable	Major	UCS_Blade_Avlblty
fltProcessorUnitInoperable	Major	UCS_Blade_Avlblty
fltProcessorUnitThermalThresholdCritical	Major	UCS_Blade_Avlblty
fltProcessorUnitThermalThresholdNonRecoverable	Critical	UCS_Blade_Avlblty
fltProcessorUnitVoltageThresholdCritical	Major	UCS_Blade_Avlblty
fltProcessorUnitVoltageThresholdNonRecoverable	Critical	UCS_Blade_Avlblty
fltMemoryUnitInoperable	Major	UCS_Blade_Avlblty
fltMemoryUnitThermalThresholdCritical	Major	UCS_Blade_Avlblty
fltMemoryUnitThermalThresholdNonRecoverable	Critical	UCS_Blade_Avlblty
fltMemoryArrayVoltageThresholdCritical	Major	UCS_Blade_Avlblty
fltMemoryArrayVoltageThresholdNonRecoverable	Critical	UCS_Blade_Avlblty
fltMemoryBufferUnitThermalThresholdCritical	Major	UCS_Blade_Avlblty
fltMemoryBufferUnitThermalThresholdNonRecoverable	Critical	UCS_Blade_Avlblty
fltComputeBoardCmosVoltageThresholdCritical	Minor	UCS_Blade_Avlblty
fltComputeBoardCmosVoltageThresholdNonRecoverable	Major	UCS_Blade_Avlblty
fltComputePhysicalPost-failure	Major	UCS_Blade_Avlblty
fltComputeIOHubThermalThresholdCritical	Major	UCS_Blade_Avlblty

Table A-8 UCSM Events—Processed for Root Cause (continued)

Event	Domain Manager Severity	EventTypeID
fltComputeIOHubThermalThresholdNonRecoverable	Critical	UCS_Blade_Avlblty
fltEquipmentChassisInoperable	Critical	UCS_Chassis_Fault
fltEquipmentChassisIdentity	Critical	UCS_Chassis_Fault
fltEquipmentChassisPowerProblem (chassis fails to satisfy power requirements in policy or at least one of power supplies failed)	Major	UCS_Chassis_Avlblty
fltEquipmentChassisThermalThresholdCritical	Major	UCS_Chassis_Avlblty
fltEquipmentChassisThermalThresholdNonRecoverable	Critical	UCS_Chassis_Fault
fltEquipmentChassisSeeprom-inoperable	Critical	UCS_Chassis_Fault
fltDcxVcDown	Major	UCS_BladeLinks
fltDcxVcMgmt-vif-down	Major	UCS_BladeLinks

Table A-9 UCSM Events—Enriched and Normalized

Event	Domain Manager Severity	EventTypeID
fltEquipmentPsuPowerSupplyProblem	Major	UCS_PowerSupply_Degradation
fltEquipmentPsuInoperable	Major	UCS_PowerSupply_Fault
fltEquipmentPsuThermalThresholdCritical	Major	UCS_PowerSupply_Degradation
fltEquipmentPsuThermalThresholdNonRecoverable	Critical	UCS_PowerSupply_Fault
fltEquipmentPsuVoltageThresholdCritical	Major	UCS_PowerSupply_Degradation
fltEquipmentPsuVoltageThresholdNonRecoverable	Critical	UCS_PowerSupply_Fault
fltEquipmentPsuPerfThresholdCritical	Major	UCS_PowerSupply_Degradation
fltEquipmentPsuPerfThresholdNonRecoverable	Critical	UCS_PowerSupply_Fault
fltEquipmentPsuIdentity	Critical	UCS_PowerSupply_Fault
fltEquipmentFanInoperable	Major	UCS_Fan_Fault
fltEquipmentFanModuleThermalThresholdCritical	Major	UCS_Fan_Degradation

Table A-9 UCSM Events—Enriched and Normalized (continued)

Event	Domain Manager Severity	EventTypeID
fltEquipmentFanModuleThermalThresholdNonRecoverable	Critical	UCS_Fan_Fault
fltEquipmentFanModuleIdentity	Critical	UCS_Fan_Fault
fltEquipmentFanPerfThresholdLowerNonRecoverable	Critical	UCS_Fan_Fault
fltEquipmentFanModuleInoperable	Major	UCS_Fan_Fault
fltEquipmentIOCardRemoved	Critical	UCS_I/OMod_Fault
fltEquipmentIOCardThermalProblem	Major	UCS_I/OMod_Degradation
fltEquipmentIOCardIdentity	Critical	UCS_I/OMod_Fault
fltEquipmentIOCardInaccessible	Critical	UCS_I/OMod_Fault
fltEquipmentIOCardPost-failure	Major	UCS_I/OMod_Degradation
fltEquipmentIOCardThermalThresholdCritical	Major	UCS_I/OMod_Degradation
fltEquipmentIOCardThermalThresholdNonRecoverable	Critical	UCS_I/OMod_Fault
fltEquipmentFexIdentity	Critical	UCS_I/OMod_Fault
fltEquipmentFexPost-failure	Major	UCS_I/OMod_Degradation
fltNetworkElementInoperable	Critical	UCS_6100_Avlblty
fltExtmgmtIfMgmtifdown	Major	UCS_Mgmt_Node
fltStorageItemCapacityWarning (used to store loging and statistics and system images; raised when usage is above 90%)	Major	UCS_Mgmt_Node
fltMgmtEntityDegraded (one of two links between two 6100s down)	Major	UCS_Mgmt_Link
fltMgmtEntityDown (both links between two 6100s down)	Critical	UCS_Mgmt_Link
fltMgmtEntityElection-failure	Critical	UCS_Mgmt_Cluster
fltMgmtEntityHa-not-ready	Major	UCS_Mgmt_Cluster
fltMgmtEntityVersion-incompatible	Critical	UCS_Mgmt_Cluster
fltMgmtEntityManagement-services-failure	Critical	UCS_Mgmt_Cluster
fltMgmtEntityManagement-services-unresponsive	Critical	UCS_Mgmt_Cluster
fltAdaptorHostIfLink-down	Major	UCS_Adapter
fltAdaptorExtIfLink-down	Major	UCS_Adapter

Table A-9 UCSM Events—Enriched and Normalized (continued)

Event	Domain Manager Severity	EventTypeID
fltPortPIoLink-down	Major	UCS_PortsLinks
fltPortPIoFailed	Major	UCS_PortsLinks
fltPortPIoHardware-failure	Major	UCS_PortsLinks
fltEtherSwitchIntFIoSatellite-connection-absent	Major	UCS_PortsLinks
fltFabricExternalPcDown	Major	UCS_PortsLinks
fltFabricFcSanPcEpDown	Major	UCS_PortsLinks
fltEquipmentIOCardFirmwareUpgrade	Major	UCS_Firmware
fltFirmwareBootUnitCantBoot	Major	UCS_Firmware
fltFirmwareUpdatableImageUnusable	Major	UCS_Firmware
fltEtherServerIntFIoHardware-failure	Major	UCS_Ethernet
fltFabricEthEstcPcEpDown	Major	UCS_Ethernet
fltFabricEthLanPcEpDown	Major	UCS_Ethernet

Infrastructure Monitoring

This section outlines the EventTypeIDs for events originating from the domain manager Infrastructure Monitoring.

Table A-10 Infrastructure Monitoring Events

Event	Domain Manager Severity	EventTypeID
KVM_ESX_Server_Unavailable	Warning	VC_Host_Avlblty
KVM_ESX_Server_Disconnected	Warning	VC_Host_Avlblty
KVM_Connection_Failure	Critical	VC_Host_Avlblty
KVM_VM_Powered_Off	—	VC_VM_Avlblty
KVM_Server_CPU_Util_High	—	VC_Host_Resources
KVM_Server_Memory_Util_High	—	VC_Host_Resources
KVM_Server_Datastore_Free_Low	—	VC_Host_Resources
KVM_Server_Disk_Reads_High	—	VC_Host_Resources
KVM_Server_Disk_Writes_High	—	VC_Host_Resources
KVM_VM_CPU_Util_High	—	VC_VM_Resources
KVM_VM_CPU_Ready_High	—	VC_VM_Resources

Table A-10 Infrastructure Monitoring Events (continued)

Event	Domain Manager Severity	EventTypeID
KVM_VM_Guest_Memory_Util_High	—	VC_VM_Resources
KVM_VM_Host_Memory_Util_High	—	VC_VM_Resources
KVM_VM_Disk_Free_Low	—	VC_VM_Resources
KVM_Cluster_Effective_CPU_Low	—	VC_Cluster_Resources
KVM_Cluster_CPU_Util_High	—	VC_Cluster_Resources
KVM_Cluster_Effective_Mem_Low	—	VC_Cluster_Resources
KVM_Cluster_Memory_Util_High	—	VC_Cluster_Resources

Service Use Cases

This appendix lists the service use cases covered as part of Prime Central for HCS 9.2.1. For more information on use case and events observed in Prime Central for HCS during common service faults, see the chapter *Understanding Common Service Faults*, in User Guide for Prime Central for Cisco Hosted Collaboration Solution 9.2.1.

Table B-1 Service Use Cases for Release 9.2.1

Name	Service Model	EventTypeId	Event Processing Stages
CUCM Critical Processes Failure	Customer Voice Service	OM_CUCM_Processes	Root cause analysis Service impact analysis
VMWare VM Failure - CUCM	Customer Voice Service	OM_CUCM_NodeRestart, OM_CUCM_OM_Connectivity, VC_VM_Avlblty	Root cause analysis Service impact analysis
VMWare ESXi Host Failure - CUCM	Customer Voice Service	OM_CUCM_NodeRestart, OM_CUCM_OM_Connectivity, VC_Host_Avlblt, VC_VM_Avlblty, UCS_BladeLinks	Root cause analysis Service impact analysis
UCS Blade Failure - CUCM	Customer Voice Service	OM_CUCM_NodeRestart, OM_CUCM_OM_Connectivity, UCS_Blade_Avlblty, VC_Host_Avlblt, VC_VM_Avlblty, UCS_BladeLinks	Root cause analysis Service impact analysis

Table B-1 Service Use Cases for Release 9.2.1 (continued)

Name	Service Model	EventTypeId	Event Processing Stages
UCS Chassis Failure - CUCM	Customer Voice Service	OM_CUCM_NodeRestart, OM_CUCM_OM_Connectivity, UCS_Chassis_Avlblty UCS_Blade_Avlblty, VC_Host_Avlblt, VC_VM_Avlblty, UCS_BladeLinks	Root cause analysis Service impact analysis
CUCM Clustering Problems (loss of SDL link, version mismatch)	Customer Voice Service	OM_CUCM_Redundancy, OM_CUCM_NodeRestart	Service impact analysis
Changes in number of registered phones	Customer Voice Service	OM_CUCM_Endpt_Connectivity, OM_CUCM_Registration	Service impact analysis
CUCxn Critical Processes Failure	Customer Voice Service	OM_CUCxn_Processes	Root cause analysis Service impact analysis
VMWare VM Failure - CUCxn	Customer Voice Service	OM_CUCxn_OM_Connectivity, VC_VM_Avlblty, UCS_BladeLinks	Root cause analysis Service impact analysis
VMWare ESXi Host Failure - CUCxn	Customer Voice Service	OM_CUCxn_OM_Connectivity VC_Host_Avlblt, VC_VM_Avlblty, UCS_BladeLinks	Root cause analysis Service impact analysis
UCS Blade Failure - CUCxn	Customer Voice Service	OM_CUCxn_OM_Connectivity, UCS_Blade_Avlblty, VC_Host_Avlblt, VC_VM_Avlblty, UCS_BladeLinks	Root cause analysis Service impact analysis

Table B-1 Service Use Cases for Release 9.2.1 (continued)

Name	Service Model	EventTypeId	Event Processing Stages
UCS Chassis Failure - CUCxn	Customer Voice Service	UCS_Chassis_Avlblty UCS_Blade_Avlblty, VC_Host_Avlblt, VC_VM_Avlblty, UCS_BladeLinks	Root cause analysis Service impact analysis
CUCxn Clustering Problems	Customer Voice Service	OM_CUCxn_Redundancy	Root cause analysis Service impact analysis
VM Resources - memory	VM IaaS	VC_VM_Resources	Service impact analysis
VM Resources - CPU	VM IaaS	VC_VM_Resources	Service impact analysis
VM Resources - CPU ready time	VM IaaS	VC_VM_Resources	Service impact analysis
VM Resources - Disk Usage	VM IaaS	VC_VM_Resources	Service impact analysis
VM Resources - Disk Latency	VM IaaS	VC_VM_Resources	Service impact analysis
CUCM Application Cold Failure	Customer Voice Service	OM_CUCM_NodeRestart, OM_CUCM_OM_Connectivity	Service impact analysis
Changes in number of registered gateways and media devices	Customer Voice Service	OM_CUCM_Endpt_Connectivity, OM_CUCM_Registration	Service impact analysis
Insufficient Virtual Memory	Customer Voice Service	OM_CUCM_App_Resources	Service impact analysis
CPU Related Problems (CPU Pegging, High CPU, etc.)	Customer Voice Service	OM_CUCM_App_Resources	Service impact analysis

Table B-1 Service Use Cases for Release 9.2.1 (continued)

Name	Service Model	EventTypeId	Event Processing Stages
Call throttling failures (code red)	Customer Voice Service	OM_CUCM_App_Resources	Service impact analysis
Call throttling failures (code yellow)	Customer Voice Service	OM_CUCM_App_Resources	Service impact analysis
High Utilization of resources shared by all customer sites (MTP, MOH, Conferencing resources)	Customer Voice Service	OM_CUCM_CC_Resources	Service impact analysis
Route List Exhausted	Customer Voice Service	OM_CUCM_CC_Resources	Service impact analysis
Media List Exhausted	Customer Voice Service	OM_CUCM_CC_Resources	Service impact analysis
CUCxn Application Cold Failure	Customer Voice Service	OM_CUCxn_OM_Connectivity,	Service impact analysis
Memory, CPU, disk threshold exceeded	Customer Voice Service	OM_CUCxn_App_Resources	Service impact analysis
Number of available licences low, expired	Customer Voice Service	OM_Voicemail_Resources	Service impact analysis
TFTP Server (for UC services) - Critical Processes Failure	Customer Voice Service	ITM_TFTP_Processes	Service impact analysis
CUP Critical Processes Failure	Customer Presence and Infrastructure Monitoring Service	OM_CUP_Processes	Event classification, Event enrichment, Service impact analysis Root cause analysis

Table B-1 Service Use Cases for Release 9.2.1 (continued)

Name	Service Model	EventTypeId	Event Processing Stages
CUP Application Cold Failure	Customer Presence and Infrastructure Monitoring Service	OM_CUP_OM_Connectivity OM_CUP_Servicability	Event classification, Event enrichment, Service impact analysis Root cause analysis
VMWare VM Failure	Customer Presence and Infrastructure Monitoring Service	OM_CUP_OM_Connectivity OM_CUP_Servicability VC_VM_Avlblty UCS_BladeLinks	Event classification, Event enrichment, Service impact analysis Root cause analysis
VMWare ESXi Host Failure	Customer Presence and Infrastructure Monitoring Service	OM_CUP_OM_Connectivity OM_CUP_Servicability VC_Host_Avlblt VC_VM_Avlblty UCS_BladeLinks	Event classification, Event enrichment, Service impact analysis Root cause analysis
UCS Blade Failure	Customer Presence and Infrastructure Monitoring Service	OM_CUP_OM_Connectivity OM_CUP_Servicability UCS_Blade_Avlblty VC_Host_Avlblt VC_VM_Avlblty UCS_BladeLinks	Event classification, Event enrichment, Service impact analysis Root cause analysis
UCS Chassis Failure	Customer Presence and Infrastructure Monitoring Service	OM_CUP_OM_Connectivity OM_CUP_Servicability UCS_Chassis_Avlblty UCS_Blade_Avlblty VC_Host_Avlblt VC_VM_Avlblty UCS_BladeLinks	Event classification, Event enrichment, Service impact analysis Root cause analysis
CUP Application Resources Degradation	Customer Presence and Infrastructure Monitoring Service	OM_CUP_App_Resources	Event classification, Event enrichment, Service impact analysis
IM Resources Exceeded	Customer Presence and Infrastructure Monitoring Service	OM_CUP_IM_Resources	Event classification, Event enrichment, Service impact analysis

Table B-1 Service Use Cases for Release 9.2.1 (continued)

Name	Service Model	EventTypeId	Event Processing Stages
AdjacencyDetached	Voice Service Signaling Offnet	OM_CUBE_SP_AdjacencyStatus	Event classification, Event enrichment, Service impact analysis Root cause analysis
MOSCQEReached CriticalThreshold	Voice Service Signaling Offnet	OM_CUBE_SP_QOS_Critical	Event classification, Event enrichment, Service impact analysis
MOSCQEReached MajorThreshold	Voice Service Signaling Offnet	OM_CUBE_SP_QOS_Major	Event classification, Event enrichment, Service impact analysis
MOSCQEReached MinorThreshold	Voice Service Signaling Offnet	OM_CUBE_SP_QOS_Minor	Event classification, Event enrichment, Service impact analysis
ServiceCardOffline Service CardStandBy	Voice Service Signaling Offnet	OM_CUBE_SP_Status	Event classification, Event enrichment, Service impact analysis
SourceAlert DynamicBlackList	Voice Service Signaling Offnet	OM_CUBE_SP_Security	Event classification, Event enrichment, Service impact analysis
CPUCongestion MemoryCongestion	Voice Service Signaling Offnet	OM_CUBE_SP_Resource	Event classification, Event enrichment, Service impact analysis
SLAViolation	Voice Service Signaling Offnet	OM_CUBE_SP_SLA	Event classification, Event enrichment, Service impact analysis

SNMP Trap Details and Example

This chapter has the list of SNMP traps sent by Cisco Unified Communications Domain Manager (CUCDM). The decoded details of an SNMP trap event is received by an SNMP trap receiver of a subcription created using the Prime Central for HCS NBI.

CUCDM Traps

The following table lists the CUCDM traps that are supported in Prime Central for HCS 9.2.1:

Table C-1 CUCDM Traps

SNMP Trap	Priority	Description
System Startup	High	VM is in ON status.
System Shutdown	High	VM is in Shutdown status.
Service Startup	High	Start the service.
Service Monitoring	High	Shutdown and then restart the service.
DR Initial Synchronization	High	DR initial synchronization is successful.
DR Database Synchronization Failure	High	Database synchronization failed.
DR File System Synchronization Failure	High	Synchronization failed for the file system.
Disk Full	Medium	Utilization has crossed the threshold.
Excessive Load	Medium	CPU utilization is greater than the threshold.
Backup	Medium	Successful or failed status is displayed.
Performance Logs	Low	Performance log did not get generated.
Health Logs	Low	Health log did not get generated.

SNMP Trap Example

```
Simple Network Management Protocol
    version: v2c (1)
    community: public
```

```
    data: sNMPv2-Trap (7)
        sNMPv2-Trap
            request-id: 1435112335
            error-status: noError (0)
            error-index: 0
            variable-bindings: 22 items
                1.3.6.1.2.1.1.3.0: 39488729
                    Object Name: 1.3.6.1.2.1.1.3.0 (iso.3.6.1.2.1.1.3.0)
                    Value (Timeticks): 39488729
                1.3.6.1.6.3.1.1.4.1.0: 1.3.6.1.4.1.1279.0.1 (iso.3.6.1.4.1.1279.0.1)
                    Object Name: 1.3.6.1.6.3.1.1.4.1.0 (iso.3.6.1.6.3.1.1.4.1.0)
                    Value (OID): 1.3.6.1.4.1.1279.0.1 (iso.3.6.1.4.1.1279.0.1)
                1.3.6.1.4.1.1279.0: 4e434f4d535f35383637
                    Object Name: 1.3.6.1.4.1.1279.0 (iso.3.6.1.4.1.1279.0)
                    Value (OctetString): 4e434f4d535f35383637
                1.3.6.1.4.1.1279.1:
506572666f726d616e6365506f6c6c696e6753746f707065...
                    Object Name: 1.3.6.1.4.1.1279.1 (iso.3.6.1.4.1.1279.1)
                    Value (OctetString):
506572666f726d616e6365506f6c6c696e6753746f707065...
                1.3.6.1.4.1.1279.2:
506572666f726d616e6365506f6c6c696e6753746f707065...
                    Object Name: 1.3.6.1.4.1.1279.2 (iso.3.6.1.4.1.1279.2)
                    Value (OctetString):
506572666f726d616e6365506f6c6c696e6753746f707065...
                1.3.6.1.4.1.1279.3:
68636d2d637573746f6d6572312d6375636d2d7075622e68...
                    Object Name: 1.3.6.1.4.1.1279.3 (iso.3.6.1.4.1.1279.3)
                    Value (OctetString):
68636d2d637573746f6d6572312d6375636d2d7075622e68...
                1.3.6.1.4.1.1279.4:
68636d2d637573746f6d6572312d6375636d2d7075622e68...
                    Object Name: 1.3.6.1.4.1.1279.4 (iso.3.6.1.4.1.1279.4)
                    Value (OctetString):
68636d2d637573746f6d6572312d6375636d2d7075622e68...
                1.3.6.1.4.1.1279.5: 3137322e32352e3132332e313937
                    Object Name: 1.3.6.1.4.1.1279.5 (iso.3.6.1.4.1.1279.5)
                    Value (OctetString): 3137322e32352e3132332e313937
                1.3.6.1.4.1.1279.6: 43303236
                    Object Name: 1.3.6.1.4.1.1279.6 (iso.3.6.1.4.1.1279.6)
                    Value (OctetString): 43303236
                1.3.6.1.4.1.1279.7: <MISSING>
                    Object Name: 1.3.6.1.4.1.1279.7 (iso.3.6.1.4.1.1279.7)
                    Value (OctetString): <MISSING>
                1.3.6.1.4.1.1279.8: 0
                    Object Name: 1.3.6.1.4.1.1279.8 (iso.3.6.1.4.1.1279.8)
                    Value (Gauge32): 0
                1.3.6.1.4.1.1279.9: <MISSING>
                    Object Name: 1.3.6.1.4.1.1279.9 (iso.3.6.1.4.1.1279.9)
                    Value (OctetString): <MISSING>
                1.3.6.1.4.1.1279.10: 5
                    Object Name: 1.3.6.1.4.1.1279.10 (iso.3.6.1.4.1.1279.10)
                    Value (Gauge32): 5
                1.3.6.1.4.1.1279.11: 437269746963616c283329
                    Object Name: 1.3.6.1.4.1.1279.11 (iso.3.6.1.4.1.1279.11)
                    Value (OctetString): 437269746963616c283329
                1.3.6.1.4.1.1279.12: 4f4d5f4355434d5f4f4d5f436f6e6e6563746976697479
                    Object Name: 1.3.6.1.4.1.1279.12 (iso.3.6.1.4.1.1279.12)
                    Value (OctetString):
4f4d5f4355434d5f4f4d5f436f6e6e6563746976697479
                1.3.6.1.4.1.1279.13: <MISSING>
                    Object Name: 1.3.6.1.4.1.1279.13 (iso.3.6.1.4.1.1279.13)
                    Value (OctetString): <MISSING>
                1.3.6.1.4.1.1279.14: <MISSING>
```

```
                    Object Name: 1.3.6.1.4.1.1279.14 (iso.3.6.1.4.1.1279.14)
                    Value (OctetString): <MISSING>
                1.3.6.1.4.1.1279.15: <MISSING>
                    Object Name: 1.3.6.1.4.1.1279.15 (iso.3.6.1.4.1.1279.15)
                    Value (OctetString): <MISSING>
                1.3.6.1.4.1.1279.16:
68747470733a2f2f6863736c61622d70632d30313a383434...
                    Object Name: 1.3.6.1.4.1.1279.16 (iso.3.6.1.4.1.1279.16)
                    Value (OctetString):
68747470733a2f2f6863736c61622d70632d30313a383434...
                1.3.6.1.4.1.1279.17: 1
                    Object Name: 1.3.6.1.4.1.1279.17 (iso.3.6.1.4.1.1279.17)
                    Value (Gauge32): 1
                1.3.6.1.4.1.1279.18: 1
                    Object Name: 1.3.6.1.4.1.1279.18 (iso.3.6.1.4.1.1279.18)
                    Value (Gauge32): 1
                1.3.6.1.4.1.1279.19: 1329314986
                    Object Name: 1.3.6.1.4.1.1279.19 (iso.3.6.1.4.1.1279.19)
                    Value (Gauge32): 1329314986
```