Oracle® Communications Diameter Signaling Router

Service Capability Exposure Function User's Guide

Release 8.5.0.2.0
F27863-04
October 2022

Oracle Communications Diameter Signaling Router Service Capability Exposure Function User's Guide, Release 8.5.0.2.0

F27863-04

Contents

B Error Codes

What's New In This Release

This section introduces the documentation updates for Release 8.5.0.2 in Oracle Communications Diameter Signaling Router (DSR) Service Capability Exposure Function (SCEF) User's Guide.

Table What's New in this Release

Particulars	Part Number	Published date
Formatting issue resolved for the Tables in following section: • MQTT Broker • CDR Field Properties • SCEF MMI Attributes • IPDD Statistics	F27863-04	October 2022
The following update is made in 8.5.0.2.0 release: • Removed all LwM2M references in this document.	F27863-03	January 2022
The following features were added in 8.5.0.2.0 release: • Support of External Group ID for NIDD • Support of External Group ID for ME	F27863-02	June 2021
The following sections were updated in 8.5.0.1.0 release for the part number • Updated the note in the SCEF MMI Configuration section. • Removed the `chargingEnabled` parameter from SCEF MMI Configuration and SCEF MMI Attributes sections. • Removed the `chargingFeatureList` parameter from the Table 6-10 table. • Removed `chargingFqdn` and `chargingRealm` parameters from the Table 6-11 table.	F27863-01	November 2020

List of Figures

List of Tables

1 Introduction

This chapter describes the Oracle Communications DSR Service Capability Exposure Function (SCEF) product, which interacts with, and implements controls on, Internet of Things (IoT) devices.

Machine Type Communication (MTC) is the communication between wired and wireless devices. It can enable a sensor or meter to communicate data (such as temperature, inventory level, etc.) to software at another location for its use. For example, sending the number of kilowatts of power used by an individual's home to the billing software at a utility company; or a refrigerator sending a user's smart phone information about what may be needed at the grocery store. The expansion of IP networks around the world has made machine-to-machine communication quicker and easier and it uses less power. These networks also allow new business opportunities for consumers and suppliers.

The end-to-end communications (between the user's equipment and the network), uses services provided by a 3rd Generation Partnership Project (3GPP) system, and optionally services provided by a Services Capability Server (SCS). The MTC application in the external network is typically hosted by an Application Server (AS) and may make use of an SCS for additional value-added services. The 3GPP system provides transport, subscriber management, and other communication services including various architectural enhancements motivated by, but not restricted to, MTC (for example, control plane device triggering).

The SCS connects to the 3GPP network to communicate with user equipment (UE) used for MTC and the MTC Interworking Function (MTC-IWF) and/or SCEF in the Home Public Land Mobility Network (HPLMN). The SCS offers capabilities for use by one or multiple MTC applications and the UE can host one or multiple MTC applications. The corresponding MTC applications in the external network are hosted on one or multiple ASs.

The SCEF is the key entity within the 3GPP architecture for service capability exposure that provides a means to securely expose the services and capabilities provided by 3GPP network interfaces. In certain deployments, the MTC-IWF may be co-located with SCEF in which case MTC-IWF functionality is exposed to the SCS/AS through the T8 interface (that is, the REST API). In deployments where MTC-IWF is not co-located with SCEF, interactions between MTC-IWF and SCEF are left up to the implementation.

SCEF allows services and capabilities to be securely used on 3GPP network interfaces by:

- providing a way to discover the exposed services and capabilities;
- providing access to network capabilities through homogenous network application programming interfaces (for example, network APIs) defined over the T8 interface; and
- abstracting the services from the underlying 3GPP network interfaces and protocols.

This document describes the how the configuration and administration of SCEF through a machine-to-machine interface (MMI) affects works with DSR and how various screens within DSR provide you with SCEF information.

Intended Scope and Audience

This content is intended for personnel who plan to provision SCEF.

The content does not describe how to install, update, or replace software or hardware.

Manual Organization

This content is organized as follows:

- Introduction contains general information about SCEF including an overview and logic information, the organization of this content, and how to get technical assistance.
- DSR SCEF Architecture describes how SCEF is configured within DSR.
- Configure SCEF describes how to access SCEF.
- Managed Objects describes the managed objects used to build the SCEF.
- SCEF MMI Attributes describes the MMI attributes used with the SCEF.

Understanding SCEF

DSR has been enhanced to support the capabilities of a Service Capability Exposure Function (SCEF). SCEF is a new network element that securely exposes the servers and capabilities provided by 3GPP network interfaces. Some functions included with SCEF include:

- Non-IP data delivery (NIDD) for low power devices

  Functions for NIDD are used to handle mobile originated (MO) and mobile terminated (MT) communication with UE, where the data used for the communication is considered unstructured from the Evolved Packet System (EPS) standpoint (which we refer to also as non-IP). The support of non-IP data is part of the Consumer Internet of Things (CIoT) EPS optimizations.
- Monitoring a device's state

  The Monitoring Events feature monitors specific events in the 3GPP system and makes the monitoring events information available using SCEF. It allows the identification of the 3GPP network element suitable for configuring the event, the event detection, and the event reporting to the authorized users, for example, for use by applications or logging. If an event is detected, the network can be configured to perform special actions like limit the UE access.
- Device triggering performs application-specific action including communication with the Service Capability Server (SCS)

  Device Triggering allows SCS to send information to the UE through the 3GPP network to trigger the UE to perform application-specific actions that include initiating communication with SCS for the indirect model or an AS in the network for the hybrid model. Device Triggering is required when an IP address for the UE is not available or reachable by SCS/AS.
- Enhanced Coverage Restriction Control

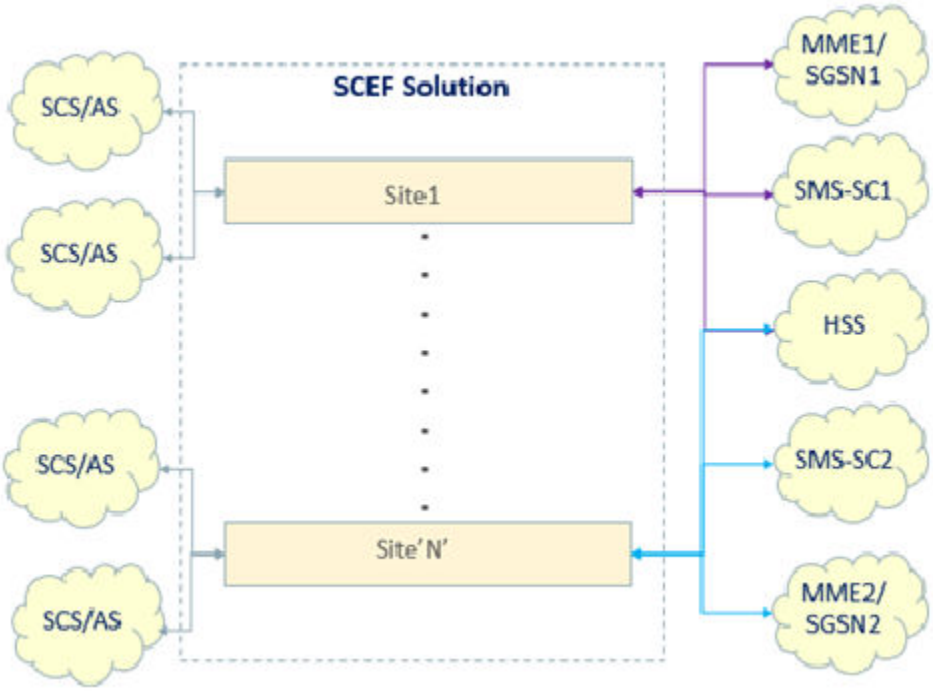
The support for Enhanced Coverage Restriction Control using SCEF enables 3rd party service providers to query status of enhanced coverage restriction, or enable/disable enhanced coverage restriction per individual UE.

- IDIH Support for SCEF
  Integrated Diameter Intelligence Hub (IDIH) support has been added for SCEF Diameter interfaces. For information about Diameter interfaces, refer to Table 1-1. The IDIH support allows users to do the following:
  - Create and manage trace filters on SCEF related Diameter interfaces to capture messages required for troubleshooting service issues
  - View traces in graphical formats
  - Filter, view, and store the results in IDIH

  For information about how to use IDIH, refer to the *IDIH User's Guide* .

The SCEF server interacts with Internet of Things (IoT) networks as a machine type communication inter-working function (MTC-IWF). Figure 1-1 shows how SCEF interacts with other DSR elements and an IoT network.

Figure 1-1 DSR SCEF Interactions



IoT devices have unique identifiers and can transmit data over a network. An IoT network can consist of numerous devices, characterized by simple design, low power consumption, brief and infrequent data transmissions, and infrequent machine transmissions (mostly they are not transmitting). The SCEF server supports IoT devices through non-IP data delivery (NIDD). An SCEF server can relay triggers from an SMS-SC function to IoT devices using Short Message Service (SMS) messages through the Diameter T4 interface. An SCEF server communicates with the home subscriber server (HSS) using the Diameter S6t and S6m interfaces. An SCEF server communicates with mobility management entity (MME) functions using the Diameter T6a and T6b interfaces. An SCEF server generates charging records and communicates with charging servers using the Diameter Ga interface. Table 1-1 provides a summary of these supported reference points.

Table 1-1 Supported Diameter Reference Points

Reference Point Name	Description
T4	Reference point used between SCEF and SMS-SC/GMSC/IWMSC
T6a	Reference point used between SCEF and serving MME
T6b	Reference point used between SCEF and serving SGSN
T8	Reference point used between SCEF and SCS/AS
S6t	Reference point used between SCEF and HSS
S6m	Reference point used between MTC-IWF and HSS

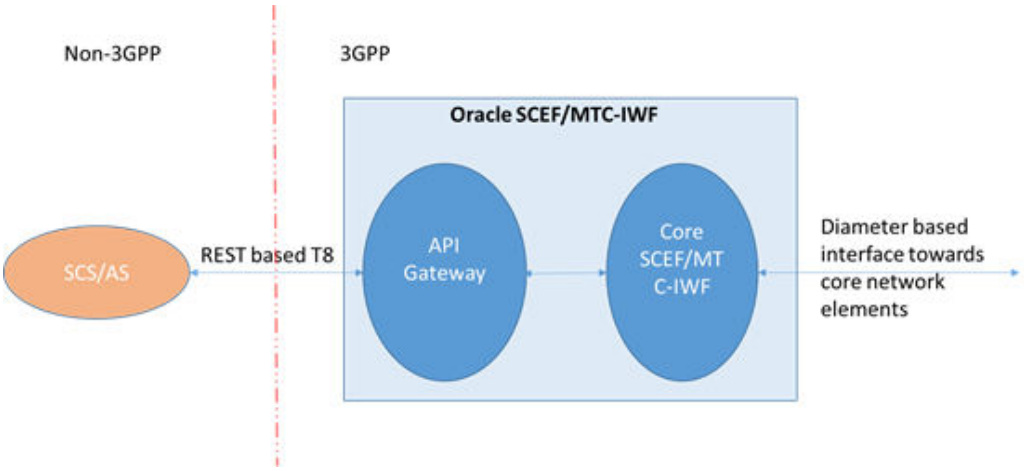
An SCEF network communicates with services capability server/application server (SCS/AS) functions using either the T8 otherwise known as the WebSocket representational state transfer (REST)ful application program interface (API) protocols using the DSR API gateway, which provides a proxy API gateway with trusted identity management, IP multimedia subsystem (IMS) access, quality of service (QoS) control, messaging services, and industry-standard security, authentication, accounting, and authorization. Configurable, extensible mechanisms support applying rate, volume, and other limits on a per-SCS/AS basis.

To support large network environments, an SCEF network can communicate with charging, home subscriber server (HSS), and mobility management entity (MME) servers using DSR.

Major Functional Components of an SCEF Network

DSR implements the functionality of both SCEF and MTC-IWF network elements. SCEF/MTC-IWF functionality at DSR can be split into two functional components as depicted in Figure 1-2:

Figure 1-2 SCEF/MTC-IWF Functionality at DSR



- The API Gateway provides the northbound interface between SCEF and Services Capability Server/Application Server (SCS/AS) based on a T8 interface. The T8

APIs are a set of RESTful APIs defining the related procedures and resources for the interaction between the SCEF and the SCS/AS.

- The Core SCEF/MTC-IWF provides southbound interface toward core network elements like HSS, MME/SGSN (Serving GPRS Support Node), and Policy and Charging Rules Function (PCRF). The following MTC call procedures are implemented at the core SCEF/MTC-IWF component with DSR acting as SCEF and MTC-IWF network elements:
  - Device triggering function (MTC-IWF functionality)
  - Non-IP data delivery
  - Monitoring event
  - Enhance coverage restriction control

API Gateway

SCEF provides a means to securely expose the services and capabilities provided by 3GPP network interfaces. The API gateway provides access to network capabilities through homogenous application programming interfaces (that is, network RESTful APIs). The SCEF API gateway provides the secured gateway functionality implementing the northbound RESTful T8 interface toward SCS/AS. DSR has implemented the following functionalities at the API Gateway:

- Northbound T8 RESTful interface
- Authentication and authorization functionality for API requests from SCS/AS:
  - Identification of the API consumer (for example, SCS/AS)
  - Profile management of SCS/AS and management of service level agreements per SCS/AS
  - Support ACL (access control list) management for individual SCS/AS
- API firewall functionality to protect from security attacks through T8 interface:
  - Protection against malformed and oversized messages received from SCS/AS
  - Whitelist of IP addresses of SCS/AS

Core SCEF/MTC-IWF

Core SCEF/MTC-IWF implements the business logic of different MTC functional call procedures specific to SCEF and MTC-IWF network elements. Core SCEF/MTC-IWF interfaces with the API gateway to send or receive the T8 requests from SCS/AS.

DSR has implemented the following MTC functional procedures of SCEF/MTC-IWF network elements.

- Non-IP Data Delivery (NIDD) provides a path to exchange unstructured data between UE and SCS/AS without requiring the user equipment (UE) to support an IP stack. Eliminating the need to support IP results in the following benefits:
  - Reduces device complexity since there is no need to support TCP/IP
  - Reduces device cost due to lower complexity
  - Reduces device power consumption due to eliminating extra messaging and overhead related to TCP/IP
  - Compatibility with older devices not supporting IP

NIDD, using the SCEF, is handled using a PDN connection to the SCEF. The UE may obtain a non-IP PDN connection to the SCEF during the attach procedure; using UE requested PDN; or using the PDP context activation procedure. An association between the SCS/AS and a PDN connection to the SCEF needs to be established to enable transfer of non-IP data between the UE and the SCS/AS. The SCEF determines the association based on provisioned policies that may be used to map an SCS/AS identity and user identity to an access point name (APN). SCEF supports both mobile terminated (MT) and mobile originated (MO) NIDD communication between UE and SCS/AS.

- The Monitoring Events feature monitors specific events in the 3GPP system and makes monitoring events information available through SCEF. This means you can identify the 3rd Generation Partnership Project (3GPP) network element suitable for configuring specific events, event detection, and event reporting to the authorized users, for example, for use by applications or logging. If such an event is detected, the network might be configured to perform special actions, for example, limit UE access.
  DSR supports the following monitoring events configuration and deletion using HSS:
  - LOSS_OF_CONNECTIVITY

    Notifies the AS when the UE loses connection and becomes offline, which signals device abnormality and need for troubleshooting.
  - UE_REACHABILITY

    Allows AS to know the status of the devices as reachable or not reachable.
  - LOCATION_REPORTING

    Allows the AS (enterprise) to track the location of the devices without GPS modules (cargo tracking).
  - CHANGE_OF_IMSI_IMEI(SV)_ASSOCIATION

    Allows AS to detect stolen devices.
  - ROAMING_STATUS

    Allows the SCS/AS to query the UE's current roaming status (the serving public land mobility network (PLMN) and/or whether the UE is in its home PLMN (HPLMN)) and notifies when that status changes.
  - UE_REACHABILITY

    Allows AS to know the status of the devices as reachable or not reachable with a status flag (idleStatusIndication flag = true).

  DSR supports both a single report event and a continuous event report for the requested monitoring events from SCS/AS. DSR supports both monitoring requests for a group of UE or single UE.
- The Enhanced Coverage Restriction Control enables 3rd party service providers (that is, SCS/AS) to query status, enhance coverage restriction, or enable/disable enhanced coverage restriction per individual UE.
- The Device Triggering feature allows the SCS/AS to deliver a specific device trigger to the UE through SCEF. The Device Trigger request is authenticated with HSS using the User Identifier received in the request. After successful authentication, SCEF forwards the Device Trigger request to the corresponding SMS-SC to be delivered to the UE.

Overview of Main Tasks

The major tasks involved with using SCEF and DSR, described in the remainder of this document, are:

- Configuring the SCEF and DSR topology
- Managing SCEF devices
- Configuring network protocols with which SCEF devices communicate
- Defining network elements with which SCEFdevices interact
- Monitoring the operation and performance of SCEF

My Oracle Support

My Oracle Support (https://support.oracle.com) is your initial point of contact for all product support and training needs. A representative at Customer Access Support can assist you with My Oracle Support registration.

Call the Customer Access Support main number at 1-800-223-1711 (toll-free in the US), or call the Oracle Support hotline for your local country from the list at http://www.oracle.com/us/support/contact/index.html. When calling, make the selections in the sequence shown below on the Support telephone menu:

1. Select **2** for New Service Request.
2. Select **3** for Hardware, Networking and Solaris Operating System Support.
3. Select one of the following options:
    - For Technical issues such as creating a new Service Request (SR), select **1**.
    - For Non-technical issues such as registration or assistance with My Oracle Support, select **2**.

You are connected to a live agent who can assist you with My Oracle Support registration and opening a support ticket.

My Oracle Support is available 24 hours a day, 7 days a week, 365 days a year.

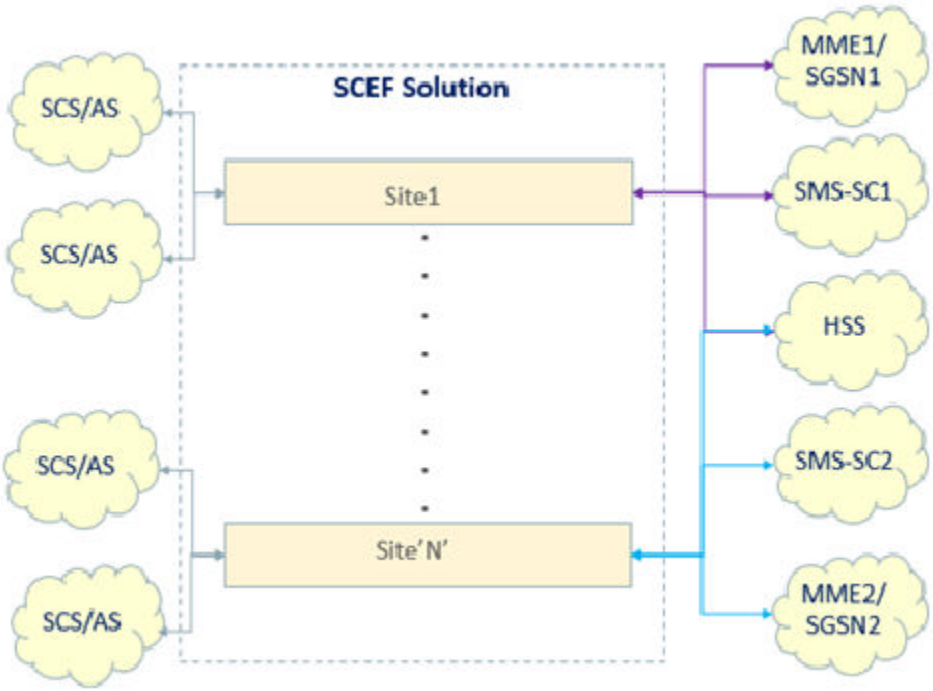
2
SCEF Functional Summary

This section provides a high-level summary of the SCEF functionality as it relates to DSR.

DSR SCEF Architecture

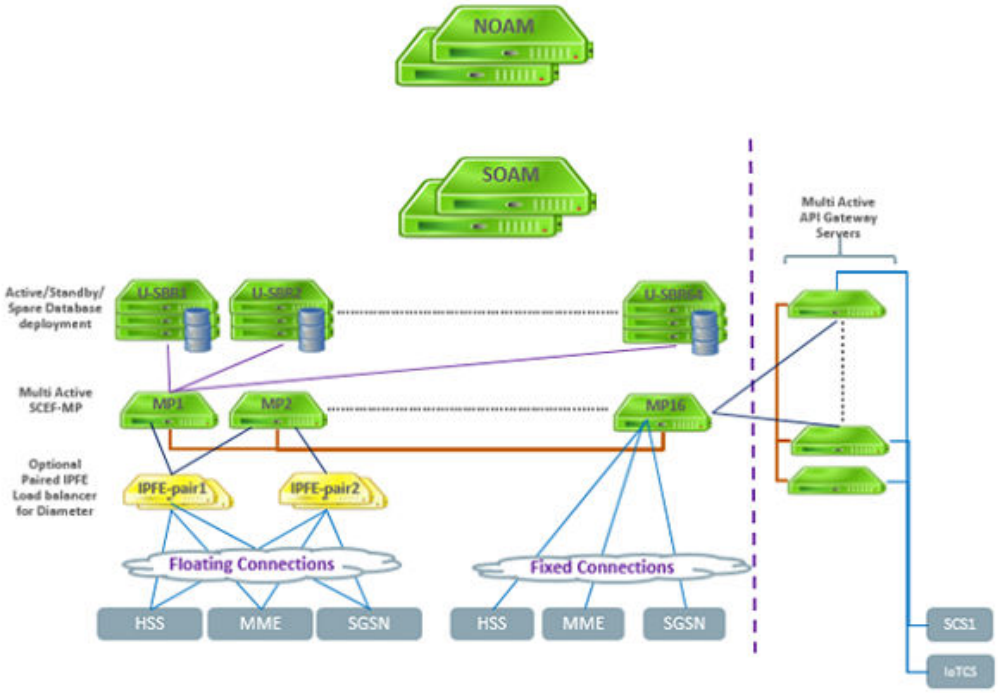
The SCEF solution, supported by a DSR network, contains one or more DSR nodes (sites). Each DSR node may be connected to 3GPP entities, like MME/SGSN, SMS-SC, and HSS, in the trusted domain; and the SCS/AS in the trusted and/or untrusted domain. The connectivity of these nodes with the DSR network is shown in Figure 2-1.

Figure 2-1 DSR-SCEF Interconnections



The DSR architecture is shown in Figure 2-2.

Figure 2-2 DSR SCEF Architecture



The solution has the following components:

- An API gateway to manage the REST interface(s) for the following:
  - Authentication of SCS/AS
  - Support for API lifecycle
  - Profile management
  - Quota and rate management
  - Load balance HTTP traffic among the DA-MP servers
- Network OAM servers deployed in active-standby redundancy model for configuration and maintenance of the DSR topology.
- Site OAM servers deployed in one, two, or three site redundancy model for provisioning of the SCEF administration data.
- IPFE servers (optional) to load balance the Diameter traffic.
- DA-MP server(s) for processing the HTTP (REST) and Diameter signaling according to the provisioning done through the site OAM servers. The DA-MPs receive the HTTP signaling traffic from the SCS/AS using the DSR API gateway application servers and the Diameter signaling traffic from the IPFE servers, if present, or from the connected Diameter peers directly. Diameter traffic generated from DA-MP servers is set to the Diameter peers directly and the HTTP traffic generated from the DA-MP servers shall be routed to the SCS/AS using the DSR API gateway.
- U-SBR server(s) deployed in one, two, or three site redundancy model for caching context data. This data is volatile, that is, the data does not persist with a server reboot, therefore, it is important to plan an adequate redundancy model.

Each SCS/AS may have a configured quota and rate for T8 messages. For example, a quota of 1000 messages in 24 hours at a rate of no more than to 100 messages per hour. Such restrictions are enforced by the DSR API gateway. If the DSR API gateway determines the rate and/or quota to be exhausted, it responds with an appropriate error message to SCS/AS. If the quota and rate are found to be within limits, the DSR API gateway forwards the T8 message to one of the DSR MP servers chosen using a simple round-robin load-sharing algorithm.

For sending a T8 request message to the SCS/AS, the DSR MP servers forward the T8 message to one of the DSR API gateway servers chosen using a simple round-robin load-sharing algorithm.

The DSR MP servers provide the SCS/AS URL in an `X-callback-url` header and provide the callback type as defined in Table 2-1 in a `X-callback-type` header to the DSR API gateway.

Table 2-1 DSR API Gateway Callback Types

X-notification-type	Notification Description
1	Monitoring Event Notification
2	Device Triggering Delivery Report
3	NIDD Uplink Data Notification
4	NIDD Downlink Data Delivery Status Notification

HTTP Message Parsing

The SCEF application parses HTTP messages as defined in 3GPP TS 29.122 specifications, T8 Reference Point for Northbound Application Programming Interfaces (APIs). The Swagger templates for the T8 messages are available on the Oracle Help Center site. Go to the latest release of the Diameter Signaling Router and then open the Service Capability Exposure Function (SCEF) YAML ZIP file.

The SCEF application receives and processes HTTP messages for Non-IP Data Delivery (NIDD), Monitoring Event, Enhanced Coverage Restriction Control, and Device Triggering APIs. The content of such messages is encoded in JSON format.

The API contained in the HTTP message is identified by a message URI prefix similar to that described in Table 2-2.

Table 2-2 Supported T8 APIs

API Name	URI Prefix
Non-IP Data Delivery	/3gpp-nidd
Monitoring Events	/3gpp-monitoring-event
Device Triggering	/3gpp-device-triggering
Enhanced Coverage Restriction Control	/3gpp-ecr-control

Database Integrity Audits

Database Integrity Audits help SCEF identify and remove alternate key records that are stale and/or pointing to invalid context records. These audits are initiated when SCEF detects that

a context record retrieved using an alternate key does not point to an appropriate context. This ability will be implemented in a future DSR release.

Non-IP Data Delivery

Functions for NIDD may be used to handle mobile originated (MO) and mobile terminated (MT) communication with UEs, where the data used for the communication is considered unstructured from the EPS standpoint (which we refer to also as Non-IP). The support of Non-IP data is part of the CIoT EPS optimizations. The Non-IP data delivery to SCS/AS is accomplished by using SCEF.

NIDD via the SCEF is handled using a PDN connection to the SCEF. The UE may obtain a Non-IP PDN connection to the SCEF either during the Attach procedure or via UE requested PDN connectivity or via PDP Context Activation Procedure.

An association between the SCS/AS and the SCEF needs to be established to enable transfer of non-IP data between the UE and the SCS/AS.

NIDD via SCEF uses the User Identity to identify which UE a particular T6a/T6b connection belongs to. The User Identity is the user's IMSI. The user's IMSI shall not be used on the interface between SCEF and SCS/AS. In order to perform NIDD configuration or to send or receive NIDD data, the SCS/AS shall use MSISDN or External Identifier to identify the user. In order to facilitate correlation of SCS/AS requests to T6a/T6b connection for a given UE, the HSS provides to the SCEF the user's IMSI, and if available, the MSISDN (when NIDD Configuration Request contains an External Identifier) or if available, External Identifier (when NIDD Configuration Request contains an MSISDN).

The NIDD procedure requested by SCS/AS is determined from the URI as described in .

Table 2-3 Supported NIDD Resources and Methods

Resource Name	Resource URI	HTTP Method(s)	HTTP Initiator
NIDD Configurations	3gpp-nidd/v1/{scsAsId}/configurations	POST	SCS/AS
Individual NIDD Configurations	3gpp-nidd/v1/{scsAsId}/configurations/{configurationId}	PUT, PATCH, GET, DELETE	SCS/AS
NIDD Downlink Data Deliveries	3gpp-nidd/v1/{scsAsId}/configurations/{configurationId}/downlink-data-deliveries	POST, GET	SCS/AS
Individual NIDD Downlink Data Deliveries	3gpp-nidd/v1/{scsAsId}/configurations/{configurationId}/downlink-data-deliveries/{downlinkDataDeliveryId}	POST, PUT, GET, DELETE	SCS/AS

Table 2-3 (Cont.) Supported NIDD Resources and Methods

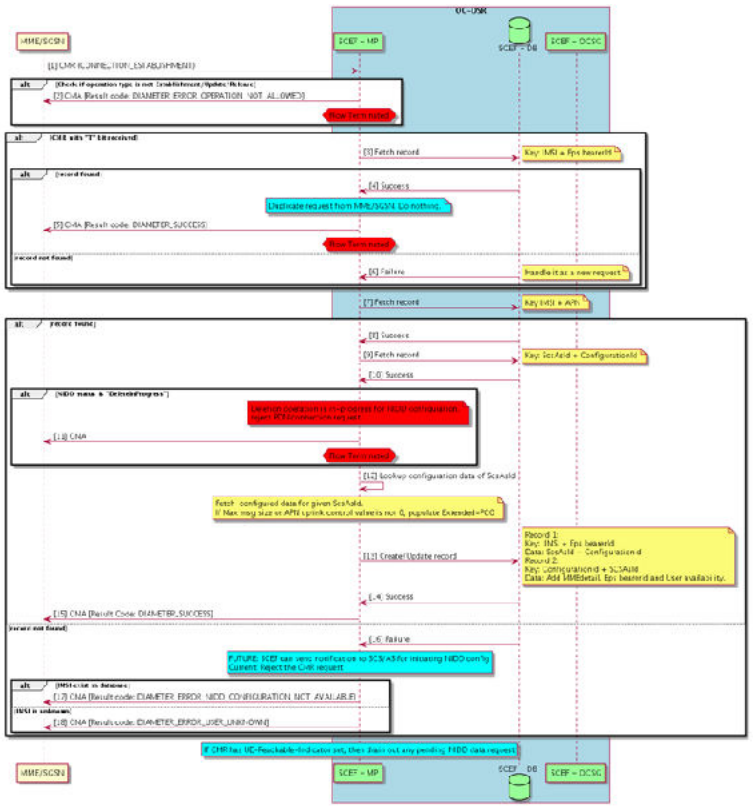
Resource Name	Resource URI	HTTP Method(s)	HTTP Initiator
NIDD Downlink Data Delivery Status Notification	{notification_destination_uri}	POST	SCEF
NIDD Uplink Data Notification	{notification_destination_uri}	POST	SCEF

PDN Connection

PDN Connection Establishment

Figure 2-3illustrates the procedure of PDN connection establishment. When the UE performs the EPS attach procedure with PDN type of Non-IP, and the subscription information corresponding to either the default APN for PDN type of Non-IP or the UE requested APN includes the **Invoke SCEF Selection** indicator, then the MME initiates a T6a/T6b connection toward the SCEF corresponding to the **SCEF ID** indicator for that APN.

Figure 2-3 PDN Connection Establishment



The MME/SGSN creates a PDN connection toward the SCEF and allocates an EPS Bearer Identity (EBI) to that PDN connection. The MME/SGSN does so by sending a Create SCEF Connection Request (User Identity, EPS Bearer Identity, SCEF ID, APN, Serving PLMN Rate Control, Serving PLMN ID, IMEISV) message toward the SCEF. If the IWK-SCEF receives

the Create SCEF Connection Request message from the MME/SGSN, it forwards it toward the SCEF.

The combination of EPS Bearer Identity, APN, and User Identity allows the SCEF to uniquely identify the PDN connection to the SCEF for a given UE. If no SCS/AS has performed the NIDD Configuration procedure with the SCEF for the User Identity, then the SCEF rejects the T6a/T6b connection setup with a cause **NIDD Configuration Not Available**.

The SCEF saves the EPS Bearer information in its Context for the user identified using User Identity and EBI. The SCEF sends a Create SCEF Connection Response (User Identity, EPS Bearer Identity, APN, PCO) message towards the MME/SGSN confirming establishment of the PDN connection to the SCEF for the UE. If the IWK-SCEF receives the Create SCEF Connection Response message from the SCEF, it forwards it toward the MME/SGSN.

PDN Connection Update

The MME/SGSN may update certain parameters that were provided in the T6a/T6b connection establishment request by sending a connection update message to SCEF. The MME/SGSN identifies the T6a/T6b connection by the IMSI and EPS Bearer Identifier. The MME/SGSN may update these parameters by a connection update message:

- Serving PLMN
- RAT Type
- Serving PLMN Rate threshold
- Origin Host and/or Origin-Realm of the MME/SGSN

The SCEF finds the context record in its database and if found updates the parameters provided in the connection update request. The SCEF then responds with the result of the update operation to the MME/SGSN.

PDN Connection Release

The MME/SGSN releases the T6a/T6b connection(s) towards the SCEF(s) corresponding to the "SCEF ID" indicator for that APN when the UE or MME/SGSN or HSS initiates a detach procedure.

The SCEF releases the T6a/b connection(s) towards the MME/SGSN corresponding to PDN connections when an NIDD Authorization Update Request from the HSS indicates that the User is no longer authorized for NIDD, or the Granted Validity Time for the NIDD configuration provided by the HSS expires or based on a NIDD configuration deletion request from the SCS/AS.

Figure 2-4 illustrates the procedure of T6a/T6b connection release when initiated by the MME/SGSN.

Figure 2-4 MME/SGSN Initiated PDN Connection Release

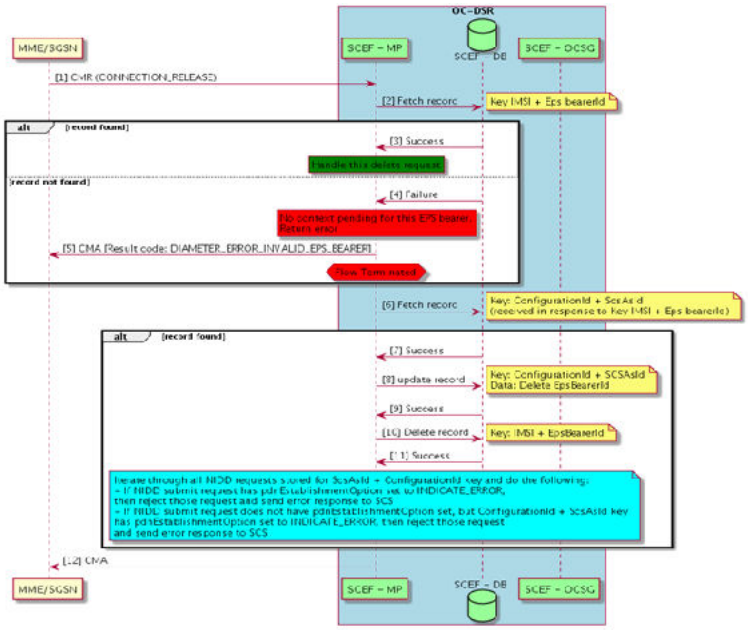
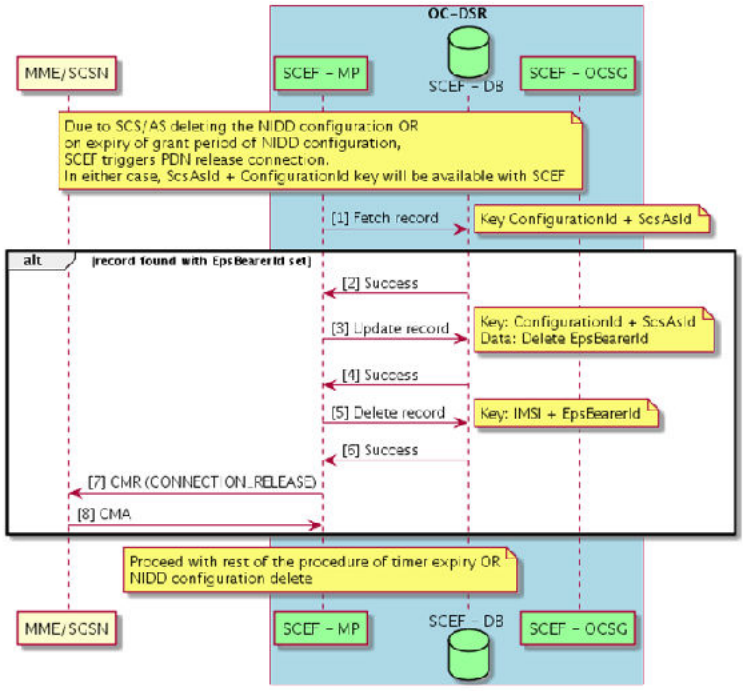


Figure 2-5 illustrates the procedure of T6a/T6b connection release when initiated by the SCEF.

Figure 2-5 SCEF-Initiated PDN Connection Release



Configuration Query by SCS Application Server

The SCS/AS may request the NIDD configuration data saved with the SCEF using an NIDD Configuration GET request. SCEF looks for the SCS/AS Identifier and the Configuration ID provided in the request and, if found, includes these parameters stored in the SCEF's database in the response.

- User Identity (External Identifier or MSISDN)
- SCS AS Identifier
- Configuration ID
- NIDD Duration
- NIDD Notification Destination Address
- List of buffered Downlink Data Delivery Packets

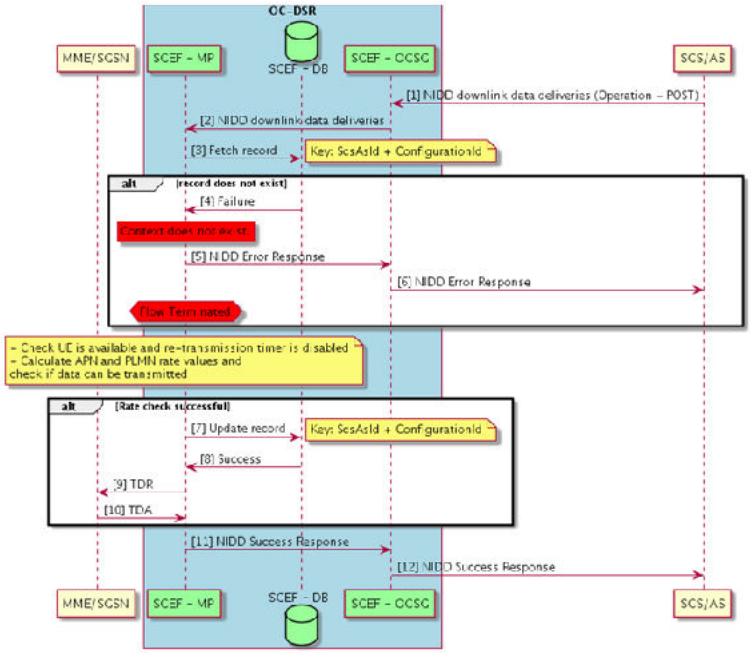
If the Configuration ID requested by the SCS/AS is not found in the SCEF's database, the SCEF responds with a 404 Not Found error.

Downlink Data Delivery

Figure 2-6 illustrates the procedure SCS/AS uses to send non-IP data to a given user as identified using the External Identifier or MSISDN.

If SCS/AS has already activated the NIDD service for a given UE, and has downlink non-IP data to send to the UE, the SCS/AS sends an NIDD Submit Request containing the External Identifier or MSISDN and the non-IP data message to the SCEF.

Figure 2-6 Downlink Data Delivery



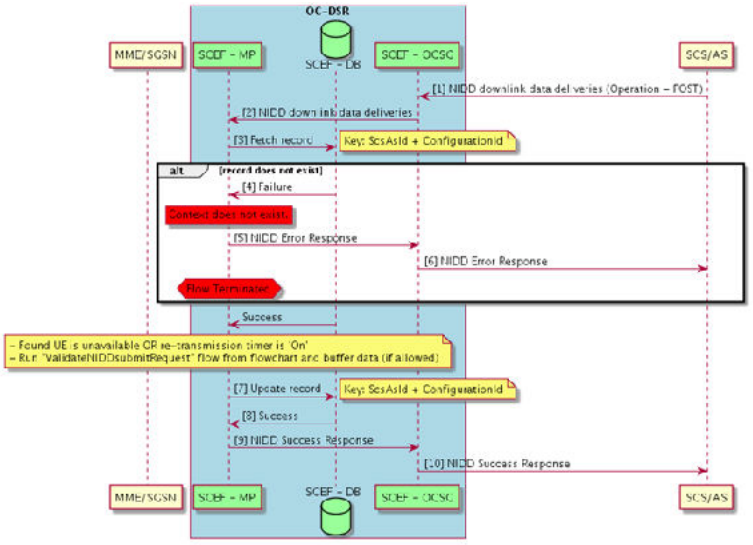
If an SCEF EPS bearer context corresponding to the External Identifier or MSISDN is found and the UE, according to the context of SCEF, is currently in a connected/reachable state, the SCEF determines whether the data delivery message rate is within the configured APN downlink rate and the Serving PLMN rate as received in the T6a/T6b connection establishment request from the MME/SGSN. If the SCEF finds the downlink data delivery message is within the rate thresholds, it attempts to send a Mobile Terminating Data message to the MME/SGSN. The SCEF also informs the MME/SGSN of the duration of time that it can wait for a response from the MME/SGSN and the duration of time up to which it can re-attempt to send the data messages. If the MME/SGSN finds the UE to be in a connected state, it attempts to deliver the data message to the UE. If the MME/SGSN cannot deliver the message within the time mentioned by the SCEF, it responds with an appropriate cause "UE temporarily not reachable." If the MME/SGSN knows when the UE is expected to be in connected state, it may inform the same to the SCEF in the Requested-Retransmission-Time parameter of the response. If the MME/SGSN is not aware when the UE may be reachable again, it stores, in its context, the SCEF Identity so that it can inform the SCEF when the UE becomes reachable.

If the SCEF does not a have an EPS bearer setup for the UE, the UE is not reachable, or the response from the MME/SGSN indicates the UE is not currently reachable, the SCEF tries to buffer the downlink data message. If the data message could be successfully buffered by SCEF, it responds with the 202 Accepted code to the SCS/AS and indicates the data is buffered. If the SCEF could not buffer the message, it indicates the cause of failure to the SCS/AS.

Data Buffering at SCEF

Figure 2-7 illustrates the procedure SCS/AS uses to send non-IP data to a given user as identified using the External Identifier or MSISDN, and the SCEF uses to decide to buffer the data message to deliver at a later point of time.

Figure 2-7 SCEF Buffering Downlink Data as UE is Not Available



Downlink data is buffered by SCEF under these conditions:

- There is no PDN connection with the MME/SGSN for the UE requested.

When there is no PDN connection with the MME/SGSN, the PDN Establishment Option is considered in the following order of preference (most preferred first) to decide whether or not to buffer the data:

  - PDN Establishment Option received from SCS/AS in the downlink data message
  - PDN Establishment Option received from SCS/AS in the NIDD configuration message
  - As configured in the NIDD Configuration Set managed object

- A previous attempt to deliver a downlink data message was responded by the MME/SGSN with a cause of "UE temporarily" not reachable.
  The UE reachability status has not been updated further by the MME/SGSN. In this case the SCEF does not attempt to send a data delivery request to MME/SGSN, rather it tries to buffer the data as soon as it receives it from SCS/AS.
- The MME/SGSN has informed the status of the UE that it is not reachable using a T6a/T6b connection establish or update request.
- The current attempt to deliver the data message was responded by the MME/SGSN with a cause of "UE temporarily" not reachable.

The following conditions need to be met for the downlink data to be buffered by SCEF:

- Data buffering must not be disabled by setting the data message lifetime to zero.
  This data duration configuration can be found in the NIDD Configuration Set managed object and has a default value of 0, that is, Data Buffering is disabled by default.
- The maximum latency of the downlink data message must be at least two times the minimum time taken for retransmitting a buffered message.
  The minimum retransmission time can be configured in the NIDD Configuration Set managed object and has a default value of 5 seconds.
- The downlink data payload size must be less than the configuration maximum packet size allowed to be buffered.
  This configuration can be found in the APN Configuration Set managed object with the default value of 100 bytes.
- There must be room to fit the downlink data message in the buffer queue for the UE.
  The length of the queue is configurable in the NIDD Configuration Set managed object with a default value of 1.

While attempting to buffer a downlink data message, assuming that all other conditions listed are found to be satisfactory, however, the queue is found to be full and the data message attempting to getting buffered has a higher priority than any message already present in the queue, the higher priority data message takes the place of the lowest priority message. A data delivery status notification shall be sent to the SCS/AS with a cause of "FAILURE" for the message that exists in the queue.

SCEF generates a Downlink Data Delivery ID for downlink data messages that are buffered, if the request does not already have it in the URI.

Any downlink data message that is buffered at SCEF resides in the data delivery queue for a maximum time as indicated by Maximum Latency attribute of the message. The maximum time is further capped by the data duration configuration parameter in the NIDD Configuration Set managed object.

Data Retransmissions

SCEF attempts to retransmit data messages that it has buffered in these scenarios:

- On expiry of a re-transmission timer that was started when the SCEF received a requested retransmission time parameter from the MME/SGSN for a data delivery request that could not be delivered by the MME/SGSN as the UE was temporarily not reachable.
- On receiving a T6a/T6b connection update message indicating the UE is now reachable.

Downlink Data Delivery Status Notification

The downlink data messages that are buffered by SCEF are either retransmitted or they expire sitting in the delivery queue. In either case, a Downlink Data Delivery Status notification is generated by SCEF and sent to the SCS/AS using the DSR API Gateway. The SCEF used the Notification Destination Address provided by the SCS/AS at the time of NIDD configuration, if provided, or the configuration in the SCS/AS managed object.

The status notification may contain one of these codes:

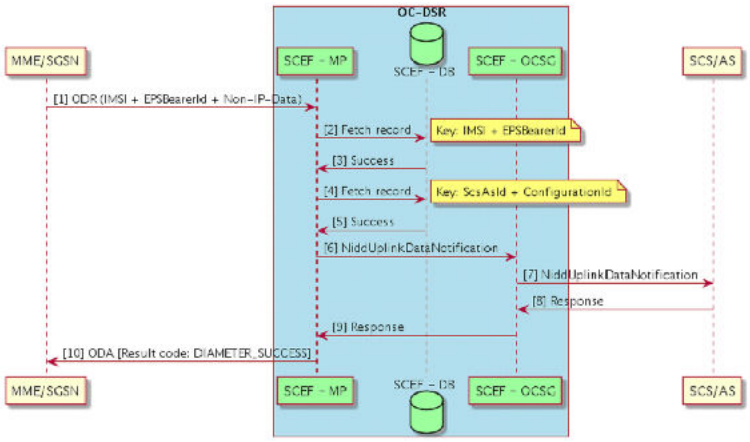
- FAILURE: When the retransmission attempt failed or the data lifetime expired.
- SUCCESS: When the data message could successfully be delivered by the MME/SGSN to the UE.

The SCEF included the Downlink Data Delivery ID of the data message in the notification for the SCS/AS to identify the same.

Uplink Data Notification

illustrates the procedure MME/SGSN uses to send non-IP uplink data to SCEF for delivery to SCS/AS.

Figure 2-8 Uplink Data Notification



The UE sends a NAS message with EPS bearer ID and non-IP data to the MME. The MME/SGSN sends the NIDD Mobile Originated Data Request containing User Identity (IMSI), EPS Bearer Identifier, and non-IP data message to SCEF. When SCEF receives the non-IP data on the T6a/T6b interface, and finds an SCEF context, it determines whether the uplink message rate is within the configured APN uplink rate. If SCEF finds the uplink data message

is within the rate thresholds, it sends the non-IP data to the appropriate SCS/AS using the Notification Destination Address provided by the SCS/AS at the time of NIDD configuration, if provided, or the configuration in the SCS/AS managed object.

> **Note:**
>
> The configured Uplink Data Rate is conveyed to the MME/SGSN and in turn to the UE in the Protocol Configuration Options IE in the T6a/T6b connection establishment answer, so it is not usually expected for the UE to send uplink data at a rate higher than that configured.

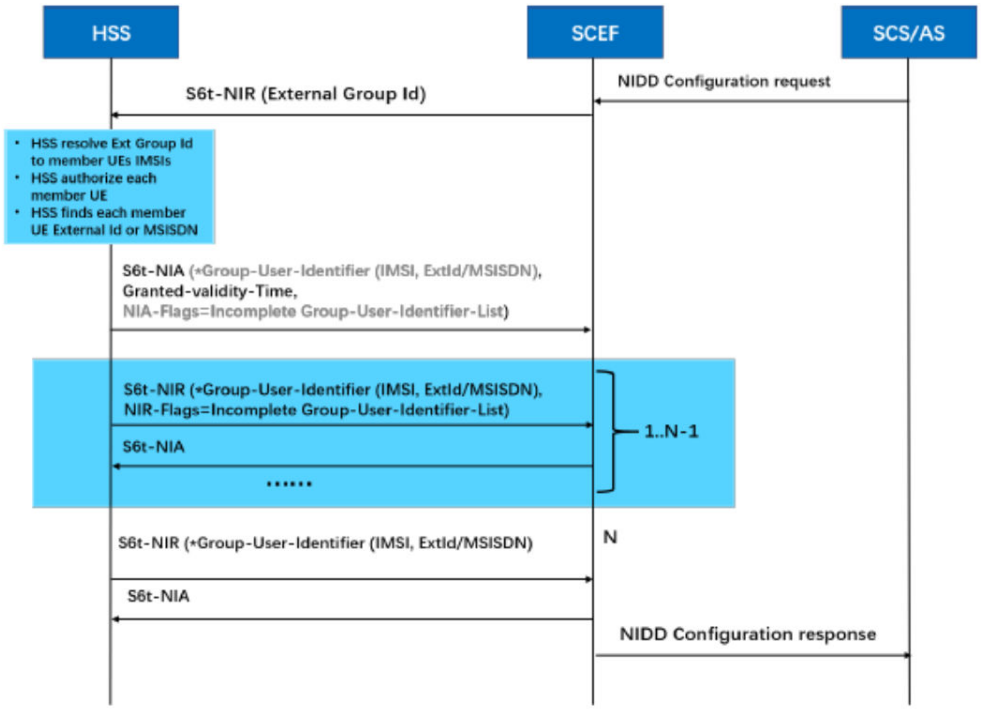
Support of External Group ID for NIDD

The group Message Delivery feature allows an SCS/AS to deliver a payload to a group of UEs. SCEF supports group message delivery for a group of UEs which are part of the same External Group ID.

The SCEF uses the SCS/AS Identifier and the External Group Identifier to determine the APN that is used to send the non-IP data to the group member UEs.

Figure 2-9 illustrates the NIDD configuration for group processing.

Figure 2-9 NIDD configuration for group processing



With the enhancement of support of external group ID, the SCEF supports the following:

- NIDD configuration with External Group Identifier for a group of UEs.
- NIDD configuration with External Group Identifier procedures as specified in 3GPP TS 23.682 Architecture enhancements to facilitate communications with packet data networks and applications.
- External Group identifier for maximum of 3K UEs.
- MT-NIDD delivery of messages to a group of UEs using External Group ID MT-NIDD delivery of messages to a group of UEs using External Group ID procedures as specified in 3GPP TS 29.122 specifications.
- If due to message size constraint, HSS determines to segment the message then SCEF shall support segmented message.
- SCEF follows the call flow as specified in Figure 2-9 for handling of the segmented message.
- SCEF aggregates the MT-NIDD group message delivery response and sends a notification to the SCS/AS.

> **Note:**
>
> Buffering is not supported in MT-NIDD group message delivery.

Reliable Data Service

Reliable Data Service (RDS) is used by the UE and SCEF when using PDN Connection of PDN Type 'Non-IP'. This service provides a mechanism for the SCEF to determine if the data was delivered to the UE and for UE to determine if the data was delivered to SCEF.

When this service is enabled, a protocol is used between the end-points of the Non-IP PDN Connection. This protocol uses a packet header to identify if the packet requires an acknowledgement or is an acknowledgment and to allow detection and elimination of duplicate PDUs at the receiving endpoint. The port numbers in the header are used to identify the application on the originator and to identify the application on the receiver.

The UE indicates its capability of supporting RDS in the Protocol Configuration Options (PCO) to SCEF or P-GW. If SCEF or P-GW supports and accepts RDS, then it indicates to UE in the PCO that RDS is used if it is enabled in the APN configuration.

With the implementation of this feature, the SCEF supports the following:

- SCEF supports Reliable data transfer for the Non-IP data delivery between UE and SCS/AS.
- SCEF supports MT NIDD Submit Request with **Reliable Data Service Configuration** option.
  When non-IP data is sent to an External Group Identifier, RDS Configuration indicates that no reliable data service acknowledgment is requested.
- SCEF suports RDS Configuration parameter in the NIDD Configuration Request.
  RDS Configuration is an optional parameter that is used to configure RDS.
- SCEF suport s updating RDS Configuration parameter in the NIDD Configuration Request update.
- SCEF includes **Reliable Data Service Indication** in the NIDD Configuration Response.
  RDS Indication indicates if the RDS is enabled in the APN configuration.

- MT NIDD data delivery supports RDS Configuration parameter.
- SCEF includes **Reliable Data Service Acknowledgement Indication** in the MT NIDD Submit Response.
  The RDS Acknowledgement Indication is used to indicate if an acknowledgment is received from UE for the MT NIDD. If RDS was requested, then MT NIDD Submit Response is sent to the SCS/AS after the acknowledgment is received from the UE. If acknowledgment is not received, then MT NIDD Submit Response is sent to SCS/AS with a cause value indicating that acknowledgment is not received.

  The UE sends RDS Acknowledgement Indication using the MO UL data.

Error Reporting

HTTP Error Reporting

The SCEF application generates an error response when an HTTP request fails to get processed successfully. The SCEF application inserts an error cause whenever possible for easy of debugging. The error cause is contained in json format for requests of type POST/PATCH/PUT. The content of the detail attribute of the problem json structure is formatted as:

```
SCEF-ERR-XXX-YYY: <human readable text description of the error>
```

where XXX is the HTTP Response Code and YYY is a 3-digit problem code Table 2-4.

Table 2-4 Problem Codes for HTTP Error Reporting

Problem Code	Problem Details
099	Generic Error
100	The configuration sets were not found for the given SCS/AS.
101	Stack-Event deserialization failed
102	An internal database error was encountered
103	A Diameter response did not contain requisite parameters to complete the transaction
104	NIDD Authorization/Grant time received from the HSS is in the past
105	A Diameter error response was received due to which the current HTTP transaction cannot be processed
106	An unexpected response was received from the Database (USBR) server
107	A Database integrity error was detected for the (IMSI, APN) Alternate Key
108	The HTTP message contains an invalid JSON content
109	The HTTP message contains a JSON content that failed schema validation
110	A context record was not found in the USBR database
111	The downlink data delivery packet was rejected as the data size exceeds the configured maximum limit

Table 2-4 (Cont.) Problem Codes for HTTP Error Reporting

Problem Code	Problem Details
112	A USBR read request failed
113	A message or event was received that was not expected in the current state of the transaction context
114	An internal error was encountered while processing an NIDD transaction
115	A PDN connection was not found for the User Entity
116	A downlink data delivery message could not be buffered because it contains a T8 Transaction Identifier that is already in use by one of the buffered data messages
117	A downlink data delivery message was rejected as it did not contain any data
118	A downlink data delivery message could not be processed as the downlink data rate limit has been reached
119	The MME was not able to deliver the downlink data to the UE
120	A parameter value in the request message is not supported (in the current version)
121	The transaction could not be processed as the operation is not valid in the current transaction state
122	Unused
123	Unused
124	The downlink data delivery message could not be buffered as the packet size exceeds the maximum allowed size for a packet that can be buffered
125	The downlink data delivery message could not be buffered as the maximum latency is too small
126	The downlink data delivery message could not be buffered as the number of currently buffered messages is at the configured maximum
127	The transaction request was failed as the UE is not authorized by Access Control
128	The transaction request as failed as the feature requested is not enabled for the requesting SCS/AS
129	The transaction request as failed as the feature requested is not enabled for the requesting UE
130	The API version requested is not supported
131	The HTTP message did not contain a mandatory parameter
132	The transaction was failed as the requesting SCS/AS is not configured

Diameter Error Reporting

Diameter error reporting problem codes will be introduced in a future DSR release.

Monitoring Event

The Monitoring Events feature monitors specific events in the 3GPP system and makes monitoring event information available using SCEF. It identifies the 3GPP network element suitable for configuring specific events, event detection, and event reporting to the authorized users, for example, for use by applications or logging. If such an event is detected, the network can be configured to perform special actions, for example, limit UE access.

The Monitoring Event procedure requested by SCS/AS is determined from the URI as described in Table 2-5

Table 2-5 Supported Monitoring Event Resources and Methods

Resource Name	Resource URI	HTTP Method(s)	HTTP Initiator
Monitoring Event Subscriptions	3gpp-monitoring-event/v1/ {scsAsId}/ subscriptions/	POST	SCS/AS
Individual Monitoring Event Subscription	3gpp-monitoring-event/v1/ {scsAsId}/ subscriptions/ {subscriptionId}	GET, DELETE	SCS/AS
Monitoring Event Notification	{notificationDestination}	POST	SCEF

Supported Monitoring Events include:

- LOSS_OF_CONNECTIVITY
- UE_REACHABILITY
- LOCATION_REPORTING
- CHANGE_OF_IMSI_IMEI_ASSOCIATION
- ROAMING_STATUS
- UE_REACHABILITY plus idleStatusIndication flag = true
- COMMUNICATION_FAILURE
- AVAILABILITY_AFTER_DDN_FAILURE plus idleStatusIndication flag = true
- NUMBER_OF_UES_PRESENT_IN_AN_AREA

Monitoring Event Subscription

To subscribe a new monitoring event configuration, the SCS/AS sends an HTTP POST message to the SCEF. The body of the HTTP POST message includes the Monitoring Type, and may include External Identifier(s) or MSISDN(s) or External Group ID, Maximum Number of Reports, Monitoring Duration, T8 Destination Address, and Group Reporting Guard Time, where the External Identifier or MSISDN indicates the

subscription for an individual UE and the External Group ID indicates a group of UEs. SCEF generates a corresponding subscription ID for a new subscription request.

> **Note:**
>
> SCEF always gives higher preference to an External Identifier when both Identifiers (External Identifier and MSISDN) are present in the Monitoring Event Configuration Request message.

The SCS/AS sends a Monitoring Subscription Request (External Identifier or MSISDN or External Group ID, Monitoring Type, Maximum Number of Reports, Monitoring Duration, T8 Destination Address, and Group Reporting Guard Time) message to the SCEF.

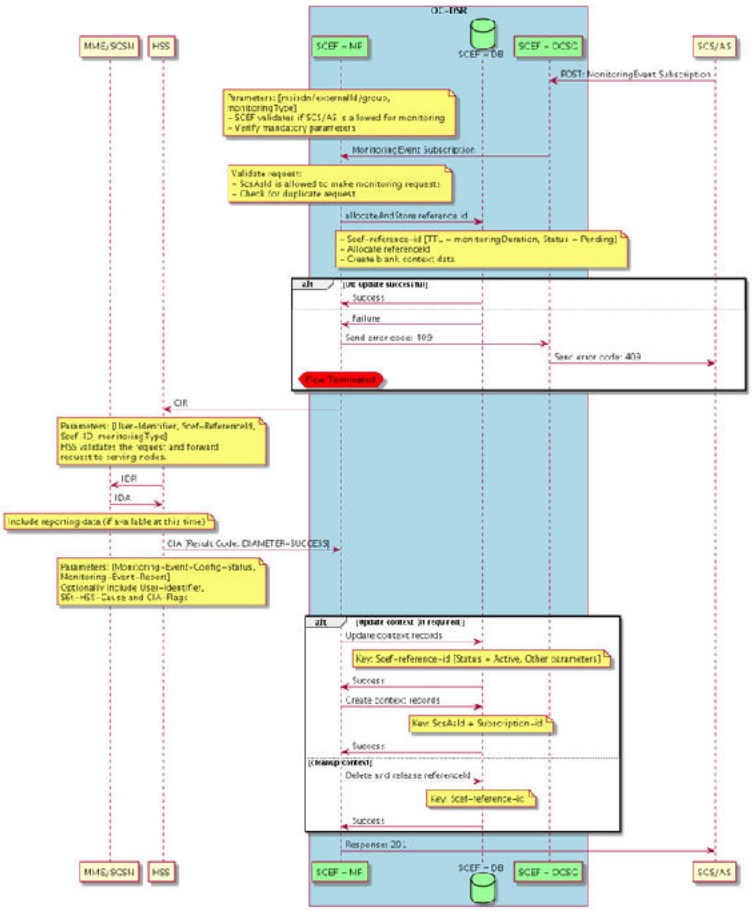
If the SCS/AS wants to configure Monitoring Event for the group of UEs, the SCS/AS can send a Monitoring Request message including External Group Identifier and Group Reporting Guard Time. A Group Reporting Guard Time is an optional parameter to indicate aggregated Monitoring Event Reporting(s), which has been detected for the UEs in a group, needs to be sent to the SCS/AS once the Group Reporting Guard Time is expired.

The SCEF stores the SCS/AS Identifier, T8 Destination Address, Monitoring Duration, and Maximum Number of Reports. The SCEF generates a subscription ID in case of a new POST request.

The SCEF sends a Monitoring Request (External Identifier or MSISDN or External Group Identifier, SCEF ID, SCEF Reference ID, Monitoring Type, Maximum Number of Reports, and Monitoring Duration) message to the HSS to configure the given Monitoring Event on the HSS in Configuration-Information-Request (CIR) message.

After processing, HSS sends a Configuration-Information-Answer (CIA) message. Then according to the result code received in the CIA message, if the result code is Success (2001), the SCEF sends a Monitoring Response (Subscription, Configuration Results, Monitoring Event Reports and Cancel Indication) message to the SCS/AS to acknowledge acceptance of the Monitoring Request; if the result code is not successful, then an error result code informs the SCS/AS about the error occurred/received.

Figure 2-10 Monitoring Event Subscription

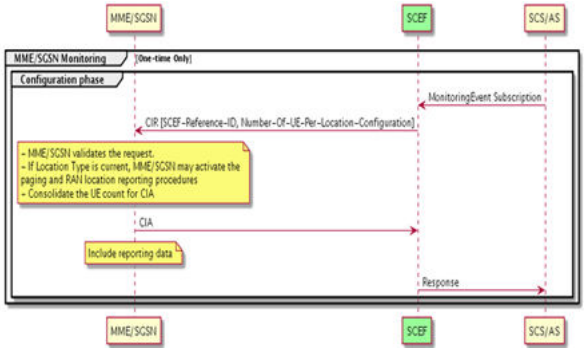


Monitoring Event Subscription Request to MME/SGSN

This procedure is used between SCEF and MME and between SCEF and SGSN to configure the monitoring events directly at the MME/SGSN through the T6a/b interface.

When the procedure is invoked by SCEF, it is used for configuring the number of UEs at a given geographic location. Each SCEF Reference ID MME/SGSN is able to process successfully, it includes the exact number of UEs known to be at the requested location in the Configuration-Information-Answer.

Figure 2-11 Monitoring Event Subscription Request to MME/SGSN



Monitoring Event Notification

Notification in Reporting-Information-Request (RIR) from HSS

This procedure is used between the HSS and the SCEF, whenever HSS needs to send a report in RIR.

When the procedure is invoked by the HSS, it is used for reporting the:

- Change of association of the UE and UICC and/or new IMSI-IMEI-SV;
- UE reachability for SMS; and
- Roaming status (Roaming or No Roaming) of the UE, and change in roaming status of the UE.

It is also used to:

- Update the SCEF with the suspend/resume/cancel status of an ongoing monitoring. Only **Cancel** is supported for current SCEF release.
- Convey reports and/or status indications for all or some UEs belonging to a group.
- Indicate the reason of communication failure.
- Indicate the information when the UE transitions into idle mode.

For group based configuration processing, if the Group Guard Timer was included in the CIR command, the HSS sends the RIR command before the Group Guard Timer expires and includes several reports and/or status indications in one or more Group-Monitoring Event Report AVPs.

> **Note:**
>
> The HSS may divide the accumulated Monitoring Configuration Indications/ immediate reports into multiple messages.
> The HSS sends immediate reports and configuration indications for the group based configuration processing using the Group-Monitoring-Event-Report.

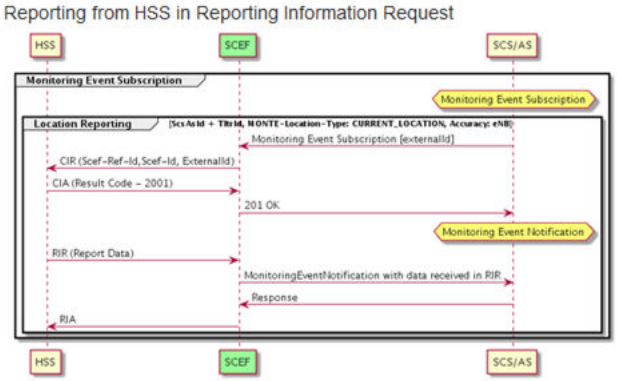
When the SCEF receives a RIR from the HSS, and at least one of the received Monitoring Event Reports has a SCEF-Reference-ID not known by the SCEF, it shall reply with

DIAMETER_ERROR_SCEF_REFERENCE_ID_UNKNOWN. In that case, if the HSS had included multiple Monitoring Event Reports in the RIR command, the SCEF includes in the Reporting Information Answer command a list of Monitoring-Event-Report-Status AVPs where the status of multiple monitoring event reports is detailed. In that AVP list, the AVPs corresponding to event reports with a successful status may be omitted by the SCEF for efficiency.

SCEF compares the Monitoring type and its value received in message with the context. If there is any mismatch, it replies with DIAMETER_ERROR_SCEF_REFERENCE_ID_UNKNOWN. Otherwise, when the SCEF receives a RIR from the HSS, the SCEF sets the Experimental-Result to DIAMETER_SUCCESS in the Reporting Information Answer and handles it according to the procedures defined in 3GPP TS 23.682 Architecture enhancements to facilitate communications with packet data networks and applications.

For each successful report data in Group-Monitoring-Event-Report and the Monitoring Event Report AVPs, SCEF sends an HTTP post notification message to SCS/AS with details as received in RIR message.

Figure 2-12 Reporting from HSS



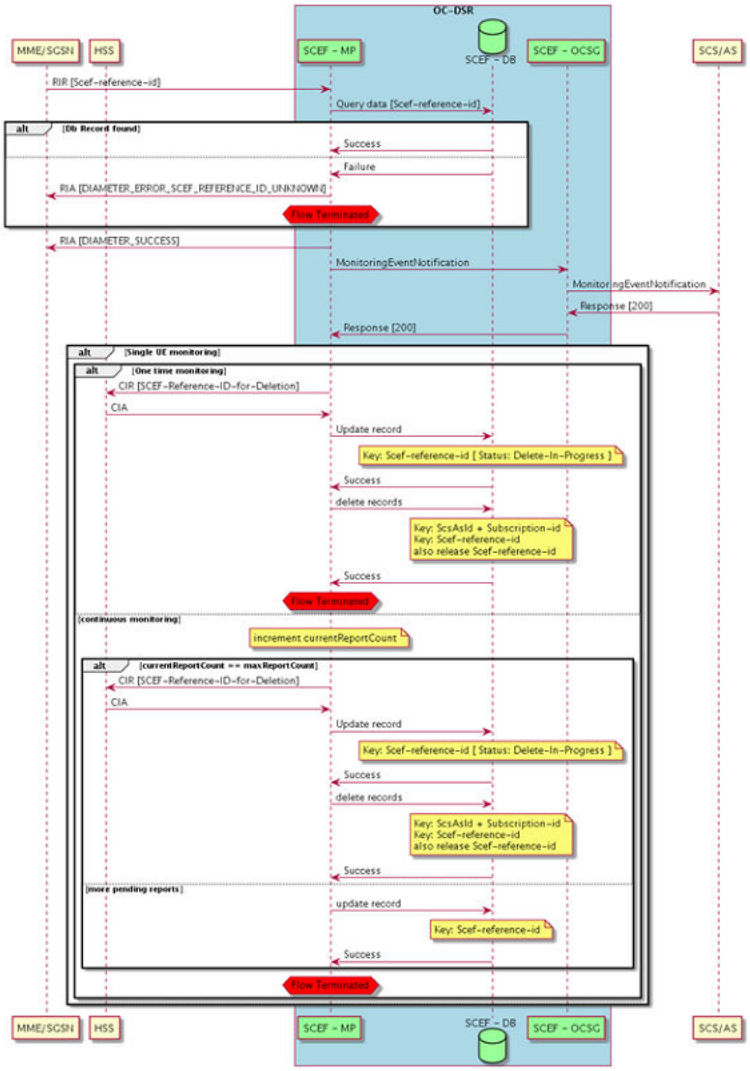
Notification in Reporting-Information-Request (RIR) from MME/SGSN

When the SCEF receives a Reporting Information Request from the MME/SGSN and at least one of the Monitoring Event Report AVPs has a SCEF-Reference-ID not known by the SCEF, it replies with Experimental-Result-Code set to DIAMETER_ERROR_SCEF_REFERENCE_ID_UNKNOWN. In that case, if the HSS had included multiple Monitoring Event Reports in the RIR command, the SCEF includes in the Reporting Information Answer command a list of Monitoring-Event-Report-Status AVPs where the status of multiple monitoring event reports is detailed. In that AVP list, the AVPs corresponding to event reports with a successful status may be omitted by the SCEF, for efficiency; otherwise, when the SCEF receives a Reporting-Information-Request command from the MME/SGSN, the SCEF sets Result-Code to DIAMETER_SUCCESS in the Reporting-Information-Answer and handles it according to the procedures defined in 3GPP TS 23.682 Architecture enhancements to facilitate communications with packet data networks and applications.

SCEF compares the Monitoring type, User Identifier, and its value received in message with the context. If there is any mismatch, it replies with DIAMETER_ERROR_SCEF_REFERENCE_ID_UNKNOWN.

For each successful report data in the Monitoring Event Report AVPs, SCEF sends an HTTP post notification message to SCS/AS with details as received in RIR message.

Figure 2-13 HTTP Post Notification



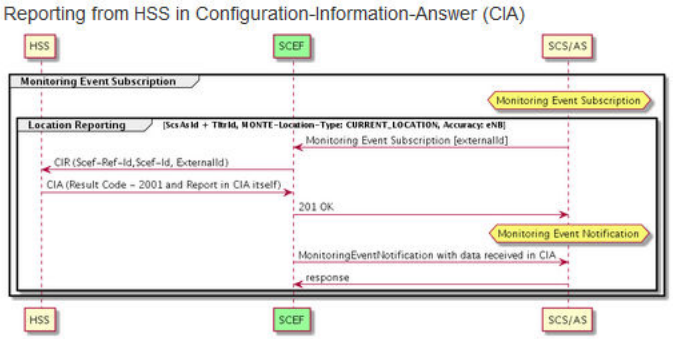
Notification in Configuration-Information-Answer (CIA)

This procedure is used between the HSS and the SCEF. HSS can send an available report for the Monitoring Event for the subscription done in the Monitoring Event Report AVPs in the Configuration-Information-Answer (CIA) message itself.

In case of a single report, for a successful report data in the Monitoring Event Report AVP, SCEF sends the report as a part of the Monitoring Subscription Response message.

In case of multiple reports, for each successful report data in the Monitoring Event Report AVPs, SCEF sends an HTTP post notification message to SCS/AS with details as received in the RIR message.

Figure 2-14 Reporting HSS



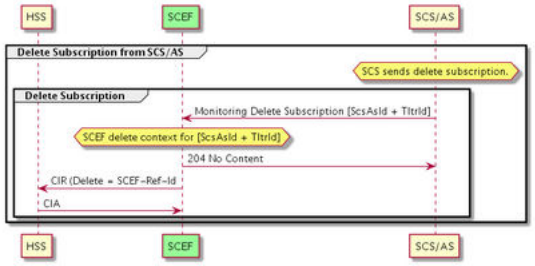
Monitoring Event Deletion Initiated from SCS/AS

SCS/AS can send HTTP message using Individual Monitoring Event Subscription and DELETE method. SCS/AS includes the subscription ID in URI, which needs to be deleted.

SCEF finds and deletes the context for Monitoring Event Subscription corresponding to SCS/AS and subscription ID received in HTTP message.

SCEF also sends Configuration-Information-Request (CIR) for deletion for SCEF Reference ID corresponding to SCS/AS and subscription ID received in the HTTP message.

Figure 2-15 Configuration-Information-Request

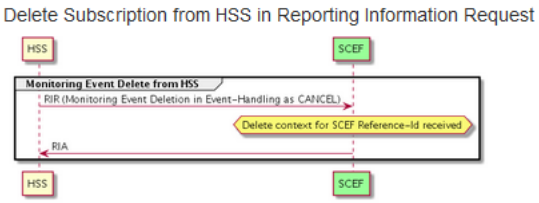


Monitoring Event Deletion Initiated from HSS

When a subscriber is deleted from the HSS while monitoring is active or the authorization for monitoring is revoked, the HSS sends an RIR command to the SCEF with the Event-Handling AVP set to the value CANCEL.

SCEF finds and deletes the context for Monitoring Event Subscription corresponding to SCEF Reference ID received in RIR message from HSS.

Figure 2-16 Delete Subscription from HSS



Monitoring Event Get

SCS/AS can send an HTTP message using the Individual Monitoring Event Subscription and GET method. SCS/AS includes the subscription ID in URI, which needs to be fetched.

SCEF finds and gets back the context data stored for the Monitoring Event Subscription corresponding to SCS/AS and subscription ID received in the HTTP message.

Support of External Group ID for ME

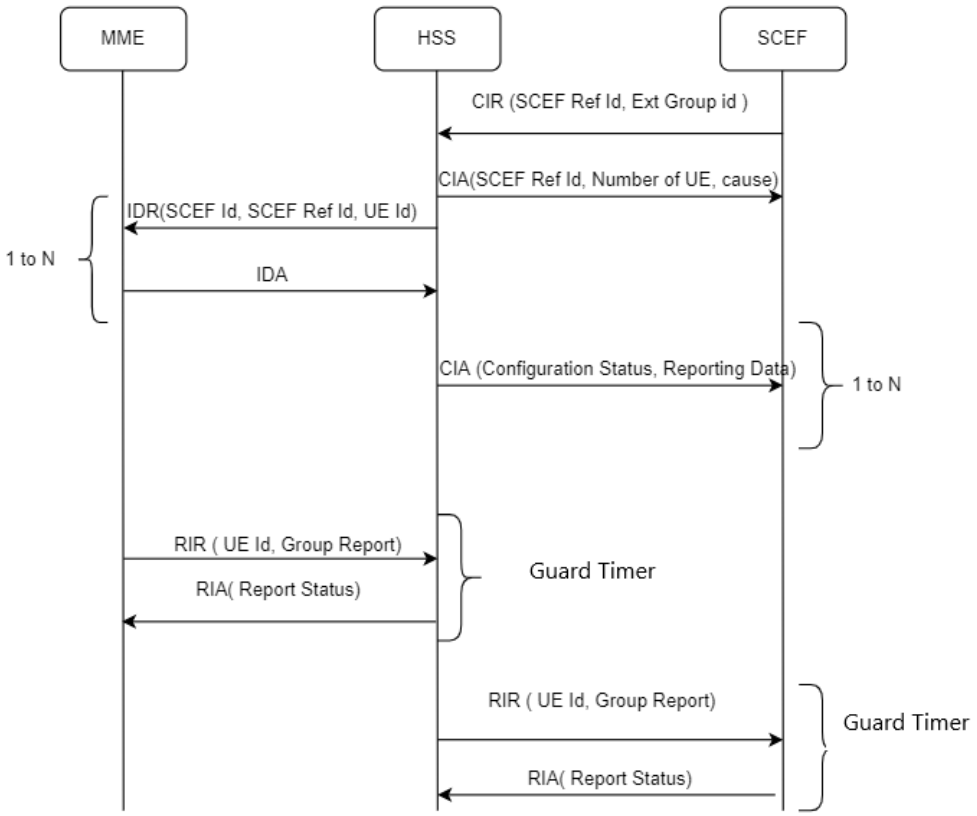
SCEF supports Monitoring Events using External Group Identifier. SCEF also support monitoring group of UE's procedure as specified in the 3GPP TS 23.682 Architecture enhancements to facilitate communications with packet data networks and applications.

SCEF supports the following monitoring event types for a group of UE's

- Loss of connectivity
- UE reachability
- Location reporting
- Change of IMSI-IMSISV association
- Roaming status
- Communication failure
- Availability after DDN failure

SCEF implements functionality to support ME reports from HSS and MME/SGSN as shown in Figure 2-17

Figure 2-17 Call flow monitoring through HSS, MME, and SGSN



With the enhancement of support of external group ID, the SCEF supports the following:

- SCEF supports both single and continuous monitoring requests from SCS/AS for external group ID.
- SCEF provides an interface to retrieve the current monitoring event configuration information which includes UE Identities (IMSI, MSISDN/External Id/External group id), ME type, Single/Continuous monitoring, and Monitoring duration.
- SCEF provides the configuration option to specify the maximum allowed monitoring number of reports and maximum allowed monitoring duration that can be requested by SCS/AS. This configuration is applied to each of the SCS/AS which is allowed to send ME configuration requests to SCEF.
- SCEF supports the ME type for a group of number of UE's at the geographical location.
- SCEF supports network-initiated explicit ME deletion procedure from HSS as specified in 3GPP TS 23.682 Architecture enhancements to facilitate communications with packet data networks and applications for a group of UE.
- SCEF supports network-initiated explicit ME deletion procedure from HSS for a group of UEs as specified in 3GPP TS 23.682 Architecture enhancements to facilitate communications with packet data networks and applications.

Guard Timer Implementation in OCSG

OCSG stores all the reports, run guard timer. After Guard timer expiry, OCSG notifies SCS/AS. When the context is deleted either by DELETE operation or context expiry, OCSG deletes the intermittent reports and stops guard timer.

Call flows

Figure 2-18 Start of Guard Timer

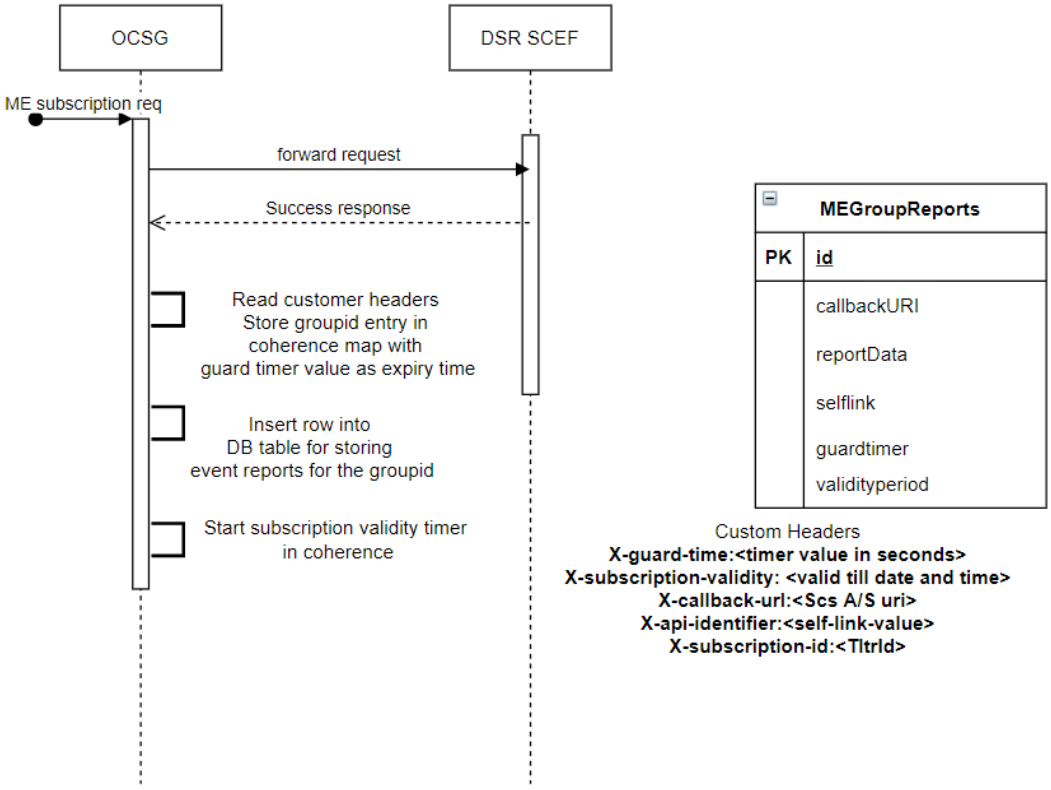


Figure 2-19 Handling event reports

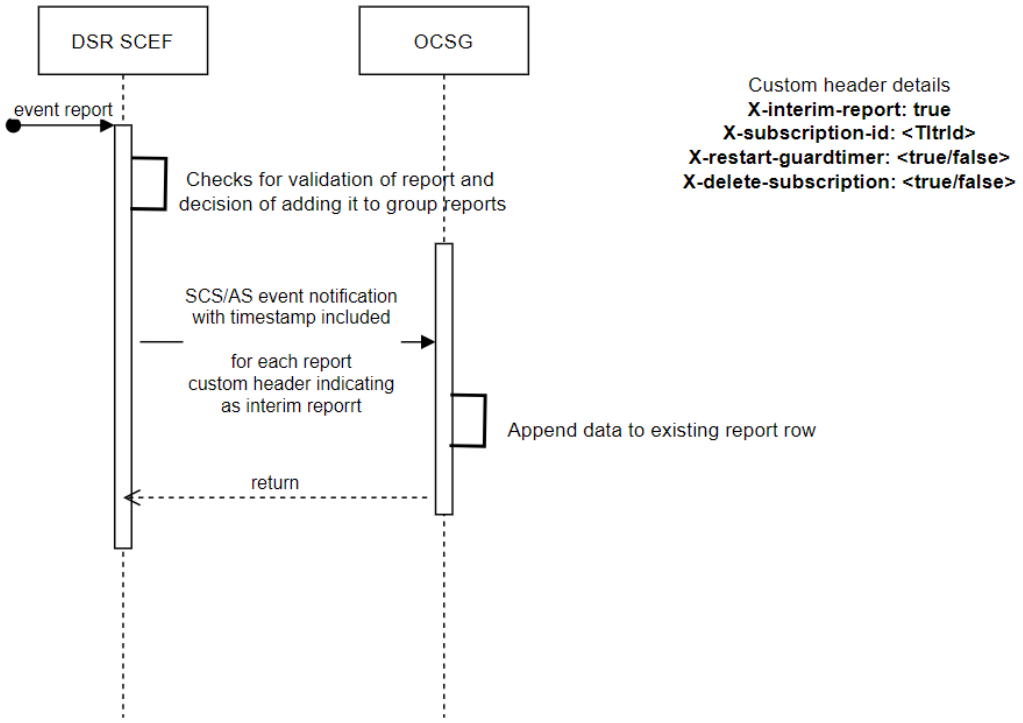


Figure 2-20 Guard timer expiry

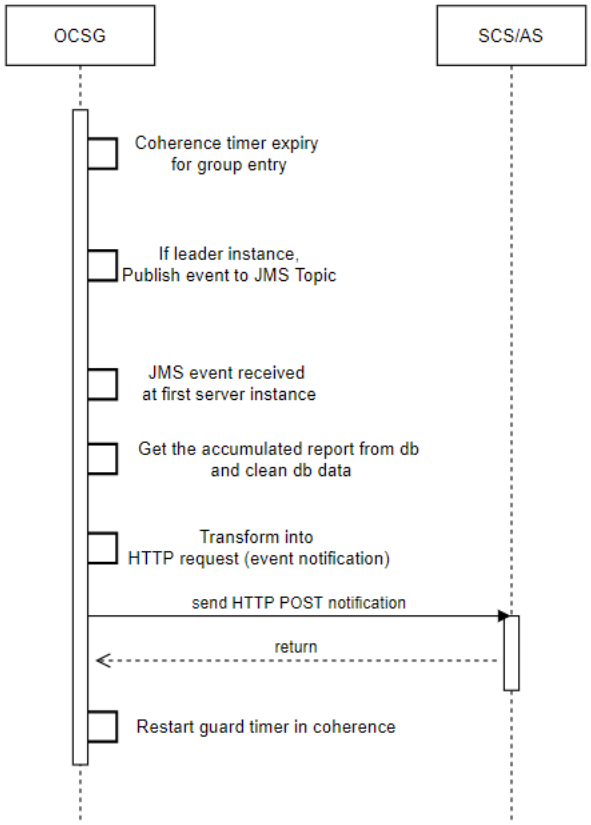


Figure 2-21 Delete Guard Timer

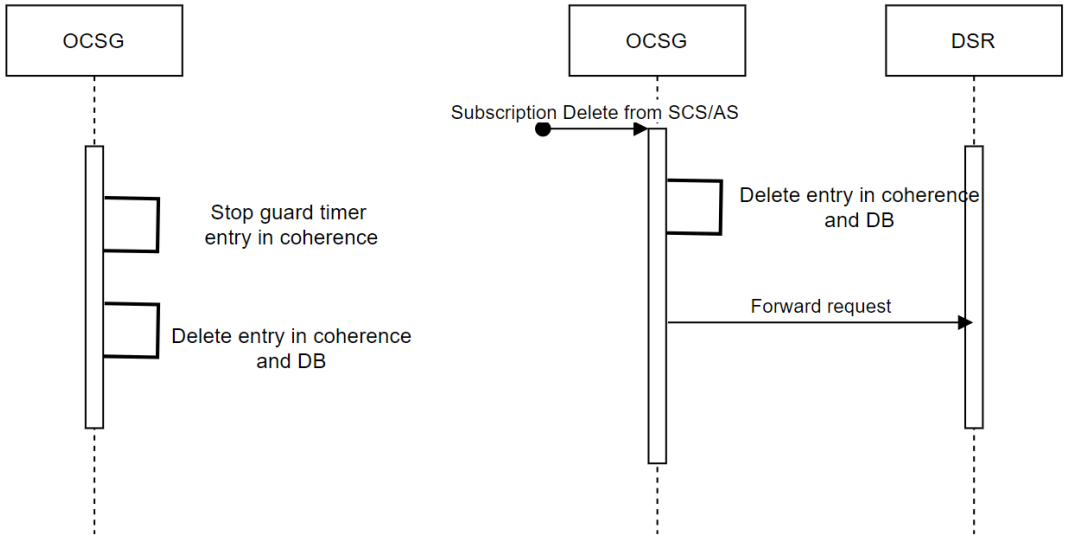
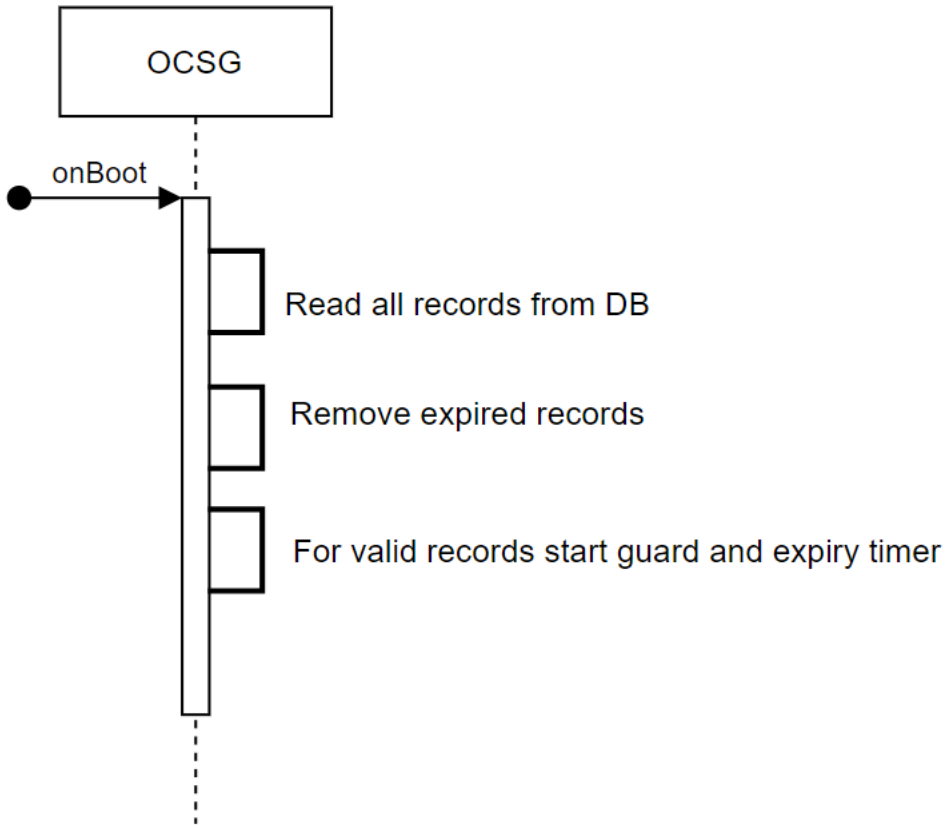


Figure 2-22 Restore Guide timer on System restore



Enhanced Coverage Restriction Control

Enhanced Coverage Restriction Control support using SCEF enables 3rd party service providers to query status of, enhanced coverage restriction, or enable/disable enhanced coverage restriction per individual UEs.

The Enhanced Coverage Restriction Control procedure requested by SCS/AS is determined from the URI as described below.

Table 2-6 Supported Enhanced Coverage Restriction Control Resources and Methods

Resource Name	Resource URI	HTTP Method(s)	HTTP Initiator
Query	3gpp-ecr-control/v1/{scsAsId}/query	POST	SCS/AS
Configure	3gpp-ecr-control/v1/{scsAsId}/configure	POST	SCS/AS

1. The SCS/AS sends an Enhanced Coverage Request (External Identifier or the MSISDN, SCS/AS Identifier, Request Type, and Enhanced Coverage Restriction Data) message to SCEF. Request Type indicates if the request needs to query the status, enable, or disable the enhanced coverage restriction. Enhanced Coverage Restriction Data provides data related to the Enhanced Coverage Restriction.

Enhanced Coverage Restriction Data is only present if the Request Type enables or disables the enhanced coverage restriction.

2. Based on operator policies, if the SCS/AS is not authorized to perform this request, or the Enhanced Coverage Request is malformed, or the SCS/AS has exceeded its quota or rate of submitting Enhanced Coverage requests, SCEF performs step 9 and provides a Cause value appropriately indicating the failure result.
3. SCEF sends an Enhanced Coverage Request (External Identifier or MSISDN Type) message to the HSS.
4. HSS examines the Enhanced Coverage Request message for the existence of an External Identifier or MSISDN, any included parameters in the acceptable range for the operator, and the Enhanced Coverage restriction by the serving MME/SGSN. If this check fails, the HSS follows step 8 and provides a Cause value indicating the reason for the failure condition to the SCEF.
   If the Request Type is to get the current status of enhanced coverage, HSS retrieves the value and follows the procedure at step 8; otherwise, if the Type is to enable or to disable the enhanced coverage, HSS sets the Enhanced Coverage Restricted parameter to the appropriate value and the procedure continues with step 5.
5. If required by the specific Enhanced Coverage Request Type and when Enhanced Coverage is supported by the serving MME/SGSN, HSS sends an Insert Subscriber Data Request (Type) message to the MME/SGSN.
6. Based on operator policies, MME/SGSN may reject the request (for example, for an overload or HSS has exceeded its quota or rate of submitting enhanced coverage requests defined by an SLA).
   The MME/SGSN updates Enhanced Coverage Restricted parameters in the MME/SGSN context.

   The MME/SGSN transfers the Enhanced Coverage Restricted parameters stored as part of its context information during the MME/SGSN change.

   > **Note:**
   >
   > UE is informed of the updated Enhanced Coverage Restricted parameters value at the next TAU/RAU or, based on the local policy, the network can detach the UE indicating re-attach is required.

7. If the Enhanced Coverage restriction is updated successfully, the MME/SGSN sends an Insert Subscriber Data Answer (Cause) message to HSS. MME/SGSN may include the Enhanced Coverage Restricted parameter in the Insert Subscriber Data Answer message.
8. HSS sends an Enhanced Coverage Response (Cause) message to SCEF. HSS includes result = success/failure and in case of success may include Enhanced Coverage Restriction Data.
9. SCEF sends an Enhanced Coverage Response (Cause, Enhanced Coverage Restriction Data) message to the SCS/AS. Cause indicates success or failure. If, in step 1, the Enhanced Coverage Request message is sent to query the status of Enhanced Coverage Restricted, then the Enhanced Coverage Restriction Data is included (in case of success) in the Enhanced Coverage Response message.

Device Triggering

The Device Triggering feature allows the SCS/AS to deliver a specific device trigger to the UE through SCEF. The Device Trigger request is authenticated with HSS using the User Identifier received in request. After successful authentication SCEF forwards the Device Trigger Request to the corresponding SMS-SC to be delivered to the UE.

The Device Triggering procedure requested by SCS/AS is determined from the URI as described in Table 2-7.

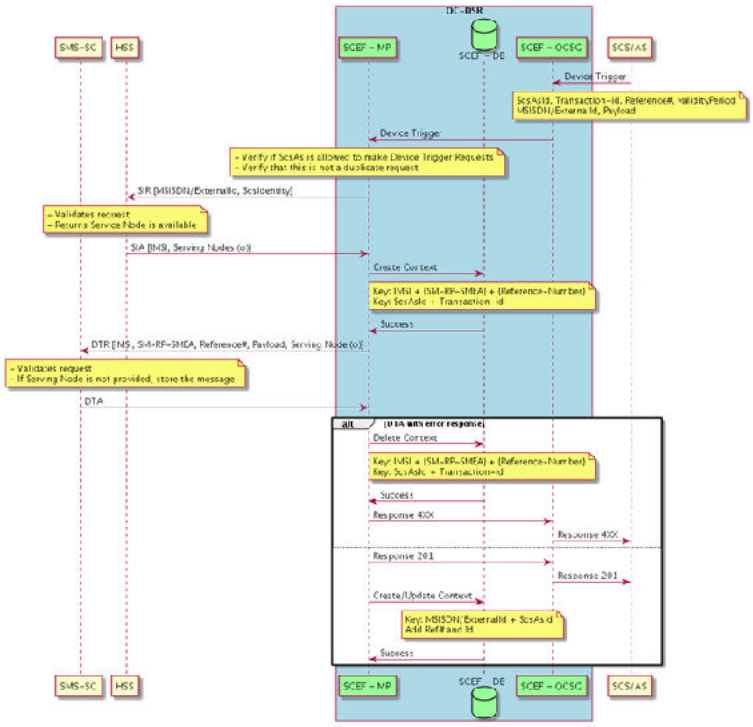
Table 2-7 Supported Device Triggering Resources and Methods

Resource Name	Resource URI	HTTP Method(s)	HTTP Initiator
Device Triggering Transactions	3gpp-device-triggering/v1/{scsAsId}/transactions	POST	SCS/AS
Individual Device Triggering Transaction	3gpp-device-triggering/v1/{scsAsId}/transactions/{transactionId}	GET	SCS/AS
Device Triggering Delivery Report Notification	{notification_uri}	POST	SCEF

Device Triggering Transaction

Figure 2-23 illustrates the procedure of creating a Device Trigger Transaction at the SCEF and SMS-SC.

Figure 2-23 Device Triggering Transaction Creation



The SCS/AS sends a Device Triggering Transaction Request (External Identifier or MSISDN, SCS/AS Identifier, Trigger Reference Number, Payload, Validity Period, Destination Address) message to the SCEF.

> **Note:**
>
> SCEF always gives higher preference to the External Identifier when both Identifiers (External Identifier and MSISDN) are present in the Device Triggering Transaction Request message.

DSR SCEF stores the External Identifier or MSISDN, SCS/AS Identifier, Destination Address, and Validity Period. If the SCS/AS is not authorized to perform this request (for example, based on Access Control policies as described in Access Control, if the SLA does not allow for it), or the Device Triggering Transaction Request is malformed, the SCEF responds appropriately indicating the error.

The SCEF sends a Subscriber Information Request (External Identifier, MSISDN, APN) message to the HSS to authorize the Device Triggering request for the received External Identifier or MSISDN, and to receive other information like IMSI, serving entities of the user, which are necessary for Device Triggering request processing.

The HSS examines the Subscriber Information Request message regarding the existence of the External Identifier or MSISDN and maps the external identifier to IMSI and/or MSISDN. If this check fails, the HSS provides a result indicating the reason for the failure condition to the SCEF.

The HSS sends a Subscriber Information Response (IMSI and MSISDN; or External Identifier, Serving Nodes, and Result) message to the SCEF to Authorize Device Triggering Request. The IMSI and, if available, the MSISDN (when Device Triggering Transaction Request contains an External Identifier) or if available, the External Identifier(s) (when Device Triggering Transaction Request contains an MSISDN) are returned by the HSS in this message.

SCEF sends a Device Trigger Request (IMSI, SME-Address, Reference Number, Payload, Validity Time, Serving Node) message to the SMS-SC to transfer the Device Trigger received from SCS/AS and identities entities serving the user. The SCEF caps the Validity Period specified by the SCS/AS at a value configured at SCEF (in the Device Triggering Configuration Set Managed Object) before sending it to SMS-SC.

The SMSC validates the identity of the user, SME-Address, and the routing information of serving entities (if available), and checks for congestion in the system. If these checks fail, then SMS-SC sends a response with result indicating the reason for failure.

The SMS-SC sends a Device Trigger Answer (Result) message to SCEF with success result if the Device Triggering Request is accepted.

The SCEF sends a Device Triggering Transaction Response message to the SCS/AS to acknowledge acceptance of the Device Triggering Transaction Request.

Transaction Query by SCS/AS

The SCS/AS may request for the Device Triggering Transaction data that is saved with SCEF using a Device Triggering Transaction GET API. SCEF looks for the SCS/AS Identifier and the Transaction ID provided in the request and if found, includes the following parameters stored in the SCEF's database in the response.

- User Identity (External Identifier or MSISDN)
- Transaction ID (only if it was received in Device Triggering Transaction Request)
- Result

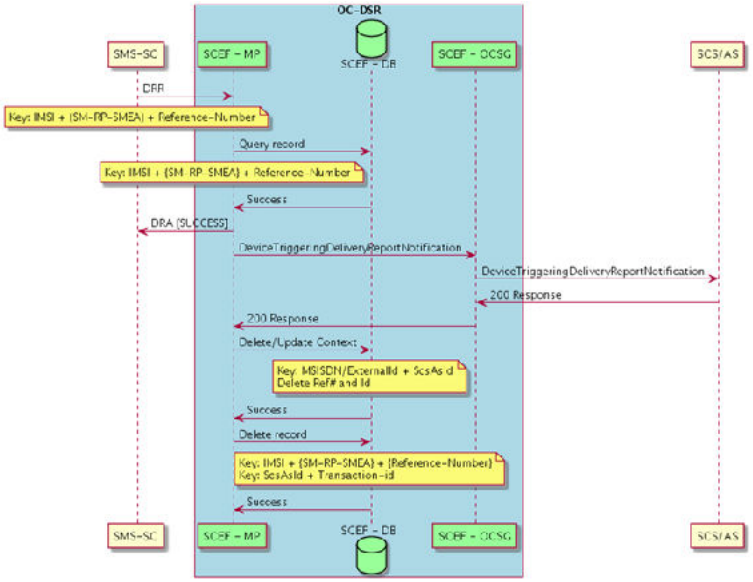
If the Transaction ID requested by the SCS/AS is not found in the SCEF's database, SCEF responds with the 404 "Not Found" error.

Device Triggering Delivery Report Notification

Figure 2-24 illustrates the procedure of sending Device Triggering Delivery Report Notification to SCS/AS.

SMS-SC sends the Device Report Request to report the success or failure of delivering the device trigger to the UE to SCEF. SCEF verifies the context for this Device Trigger exists and sends the notification to SCS/AS with an appropriate delivery result. SCEF uses the Notification Destination Address provided by SCS/AS at the time of Device Triggering Transaction, if provided, or uses the configuration in the SCS/AS managed object.

Figure 2-24 Device Triggering Delivery Report Notification



Access Control

The SCEF application provides support for multi-tenancy of SCS. This is achieved by Access Control Logic (ACL).

ACL ensures UE (IOT devices) belonging to one SCS are not accessed by another SCS.

ACL performs this functionality on HTTP requests:

- Validate SCS: If SCS is not pre-configured in SCEF, it returns a 401 Unauthorized error. If SCS is configured, but the feature (requested in message) is not enabled for SCS, then it returns a 401 Unauthorized error or it displays “Validate SCS accessibility to UE.”
- Validate SCS accessibility to UE: Extract UE-Identifier from message and validate if SCS is allowed to access the UE for the specific requested feature, if not, then it returns a 401 Unauthorized error or it allows the message for further processing.

IP Device Handler

IP Device handler manages unified traffic for Non-IP and IP devices and distributes the traffic to respective components.

This module is developed using API Gateway Actions framework. The IP Device handler is connected to the T8 API blocks in API gateway in request path (uses API gateway one push script).

IP Device handler relies on a database table to identify the device type. The database schema for the table is defined in Table 2-8

Table 2-8 IP Device Table schema

Table Name	Column Name	Data Type
scef_ip_device_info	external_id	varchar(256)
	msisdn	bigint(20)
	imsi	bigint(20)
	device_id	varchar(256)
	apn	varchar(120)
	ip	varchar(50)
	type	tinyint(4) (2- MQTT)
	active_appserver	varchar(20)
	status	varchar(20)
	aaa_session_id	varchar(256)

When the API Gateway receives a "Nidd Downlink Data Delivery" or "Monitoring Eventsubscription" request then IP device handler checks if the header "Xscef-mqtt-topic-name" is present for "Nidd Downlink Data Delivery" request and header "Xscef-message-type" is present for "Monitoring Eventsubscription" request. If these headers are not present for the corresponding requests, then those requests are Non IP requests and are forwarded to Non IP flow. If headers are present as mentioned, then IP device handler retrieves the device identifier from the request and checks it in scef_ip_device_info table. Based on the device type, the request is passed to MQTT Broker service for further processing. Refer to MQTT Broker for more information.

MQTT Broker

MQTT Broker (MB) acts as cross proxy converting the MQTT messages to HTTP messages and vice versa to enable IP device communication between IoT devices and SCS/AS. The MQTT Broker enables IoT device communication for this use case.

MQTT Broker complies with MQTT 3.1.1 specification and supports the following features:

1. MQTT Subscriber messages from the application to a specific topic and device
2. Subscribe message from the application to a specific topic
3. Publish message from the application to a specific topic and device
4. MQTT Subscribe messages from device to a specific topic
5. Publish message from device to a specific topic
6. High availability
7. Provide a rest interface to provision MQTT details into system
8. Integration with AAA server to get MQTT device IP details
9. Integration with AAA server to update the device IP details in MQTT broker
10. Route T8 messages to DSR and MQTT Broker depending on the device details provisioned in the system

11. Perform Access control checks based on configured MSISDN range and domain IDs of external identifier for MO and MT messages per application
12. Perform configured APN rate control per MQTT CONNECT, SUBSCRIBE, PUBLISH to and from devices
13. Cross-routing functionality in OCSG cluster to handle T8 requests if the devices are connected to different AppServer

MQTT Broker provides a unified T8 interface for IP device communication enabling NIDD Downlink Data Delivery (POST) and monitoring event (POST & DELETE) call flows. For more details on the required configuration, see section MQTT Configuration.

T8-MQTT-Message Mapping

This section describes the mandatory parameters requires in T8 messages and their mapping to MQTT parameters.

Table 2-9 NIDD Downlink Data Delivery (POST)

Parameter Name	Type	Mandatory/Optional	MQTT mapping
SCSASID in Request URI	Input	Mandatory	N/A
ConfigID in request URI	Defaulted to 'IPDD'	Mandatory	N/A
ExternalId in Json Body	Input	Optional	Mapped to corresponding MQTT Device ID, this field is optional if MSISDN is present or for Broadcast messages
MSISDN in Json Body	Input	Optional	Mapped to corresponding MQTT Device ID, this field is optional if ExternalID is present or for Broadcast messages
Self in Json Body	Input	Optional	Stored for DLD buffered data delivery status notifications in case of MT buffering
Data in json body	Input	Mandatory	MQTT PUBLISH Data
Custom Header "`Xscef-message-type: MQTT`"	Input	Mandatory (in case of broadcast)	N/A
Custom Header "`Xscef-mqtt-topic-name:<topic>`"	Input	Mandatory	MQTT PUBLISH topic name

Table 2-10 NIDD Buffered Message Delivery Status Notification

Parameter Name	Type	Mandatory/ Optional	MQTT Mapping
`niddDownlinkDataTransfer`	Input	Mandatory	Publishing topic name
`deliveryStatus`	Input	Mandatory	N/A
`externalId`	Input	Mandatory	To identify this notification belongs to which MT data delivery.

Table 2-11 Monitoring Event subscription (POST)

Parameter Name	Type	Mandatory/ Optional	MQTT Mapping
Custom Header "`Xscef-message-type: MQTT`"	Input	Mandatory (in case device id not present)	
`SCSASID` in Request URI	Input	Mandatory	N/A
`ExternalId` in Json Body	Input	Optional	Mapped to corresponding MQTT Device ID, this field is optional if MSISDN is present or subscriptions without device id.
`MSISDN` in Json Body	Input	Optional	Mapped to corresponding MQTT Device ID, this field is optional if External ID is present or subscriptions without device id.
Notification Destination in Json Body	Input	Mandatory	N/A
Monitoring type in Json body	Input	Mandatory	MQTT topic name
`subscriptionId`	Output	Mandatory	N/A

Table 2-12 Monitoring Event subscription (DELETE)

Parameter Name	Type	Mandatory/Optional	MQTT Mapping
Custom Header "`Xscef-message-type: MQTT`"	Input	Mandatory	
SCSASID in Request URI	Input	Mandatory	N/A
subscriptionId in Request URI	Input	Mandatory	N/A

Table 2-13 Monitoring Event subscription Notification

Parameter Name	Type	Mandatory/ Optional	MQTT Mapping
`externalIds in Json body (external Id)`	Input	Mandatory	To identify this notification belongs to which MT data delivery
`Monitoring type in Json body`	Input	Mandatory	MQTT topic name
`Subscription in Json Body (Subscription Id)`	Input	Mandatory	N/A
`Data in Json Body`	Input	Mandatory	Data in PUBLISH message from device

MQTT Call Flows

Device Subscriptions

The following call flow explains the procedure followed in device subscriptions.

Figure 2-25 Device Subscription Call Flow

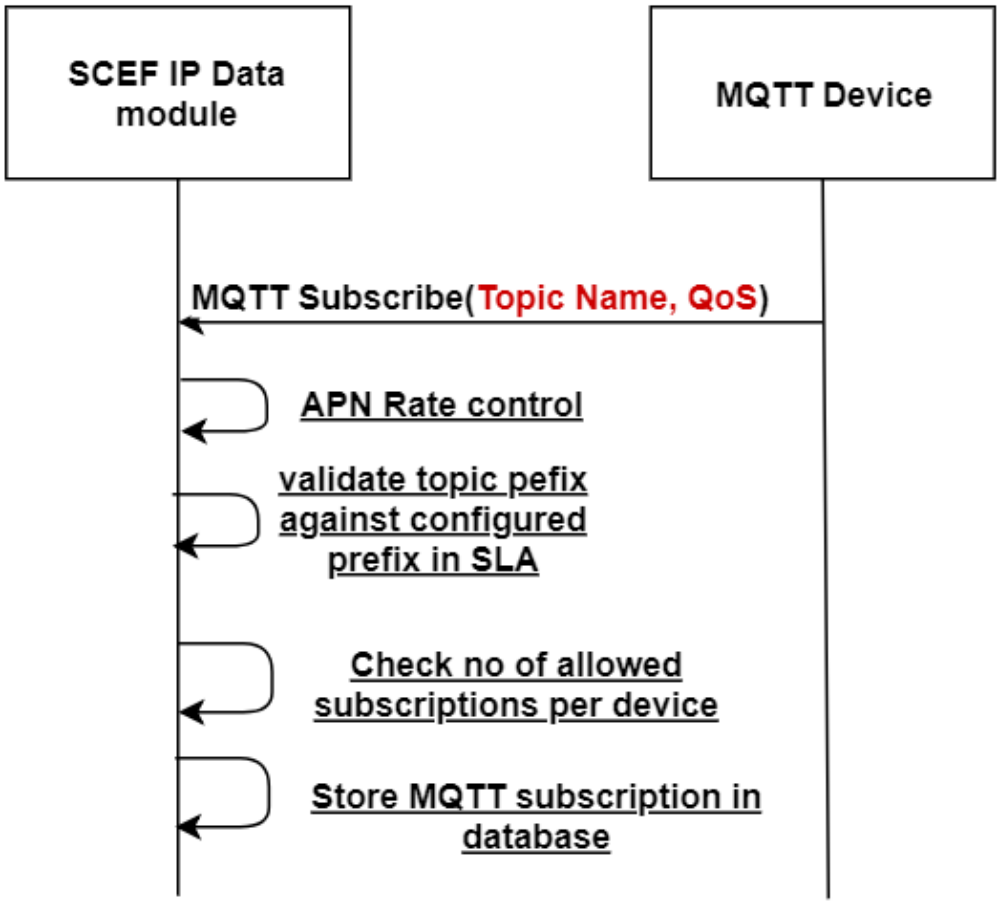


Table 2-14 Device Subscription Response Code

Response Code	Reason
MqttException (128)	• Duplicate Subscription • NoOfDeviceSubscriptionAllowed Check failed • APN Rate control check failed
Successful Subscribe Ack	Subscription is Successful

Application Subscriptions

The following call flow explains the procedure followed in application subscriptions.

Figure 2-26 Application Subscription Call Flow

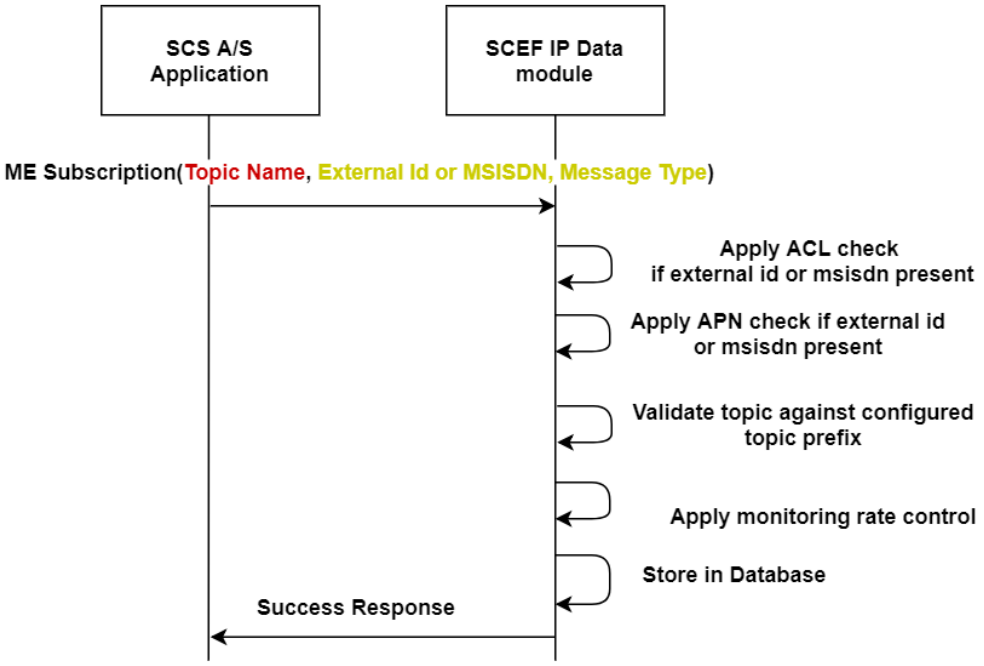


Table 2-15 Application Subscription Response Code

Response Code	Reason
200	Subscription is successful
400	• Duplicate Subscription • Bad request : Prefix Not configured in SLA • MQTT Client not configured
500	Processing error

Data Delivery (Single)

The following call flow explains the procedure followed for delivering MQTT MT messages to a single device.

Figure 2-27 Data Delivery (Single) Call Flow

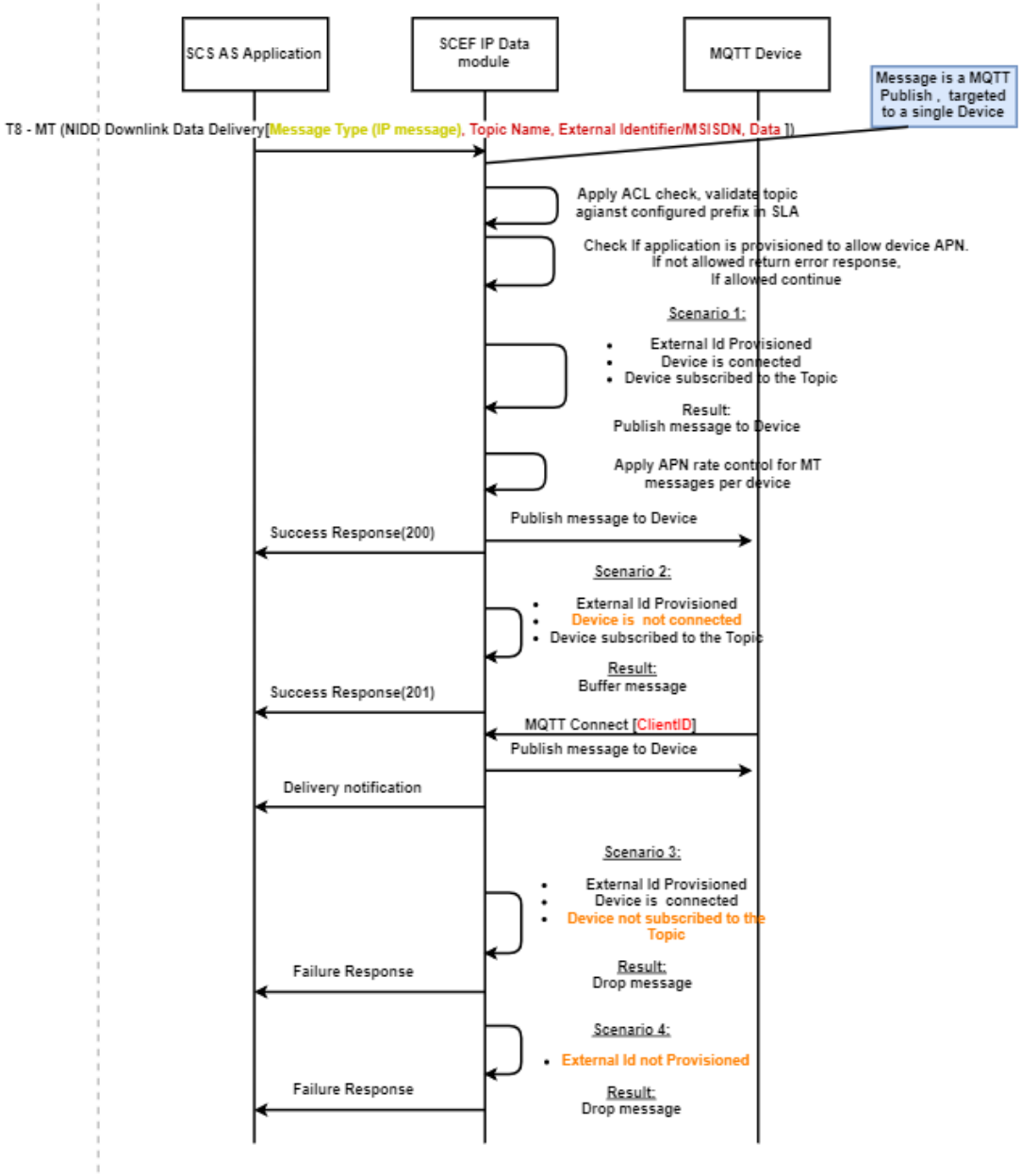


Table 2-16 Data Delivery (Single) Response Code

Response Code	Reason
200	Device is in connected state and data is delivered
201	Buffering
400	• Bad request : Prefix Not configured in SLA • MQTT Client not configured
404	Device not subscribed to topic
429	Too Many Requests for device APN rate control limit is reached

Table 2-16 (Cont.) Data Delivery (Single) Response Code

Response Code	Reason
500	Processing error

Data Delivery (Broadcast)

The following call flow explains the procedure for delivering MQTT MT message broadcast.

Figure 2-28 Data Delivery (Broadcast) Call Flow

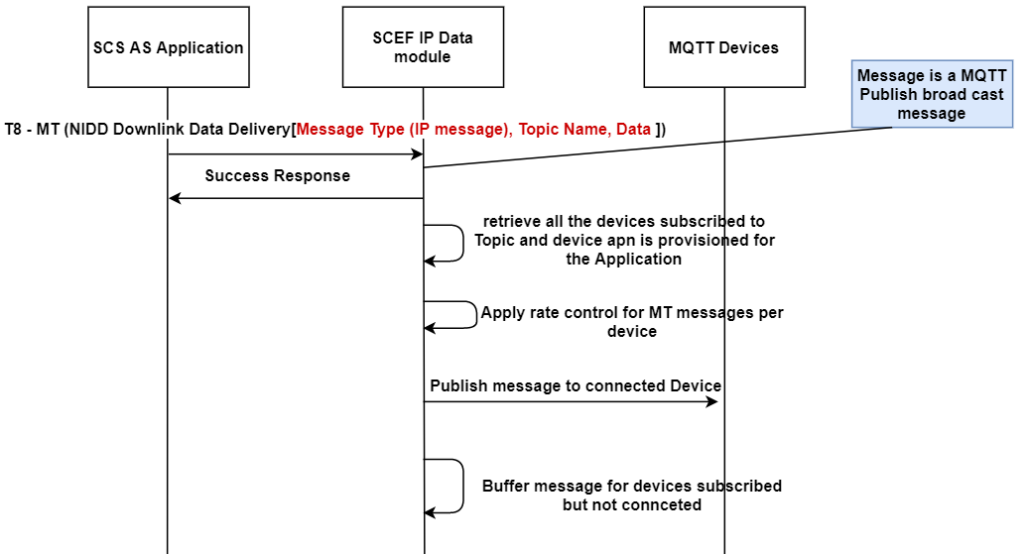


Table 2-17 Data Delivery (Broadcast) Response Code

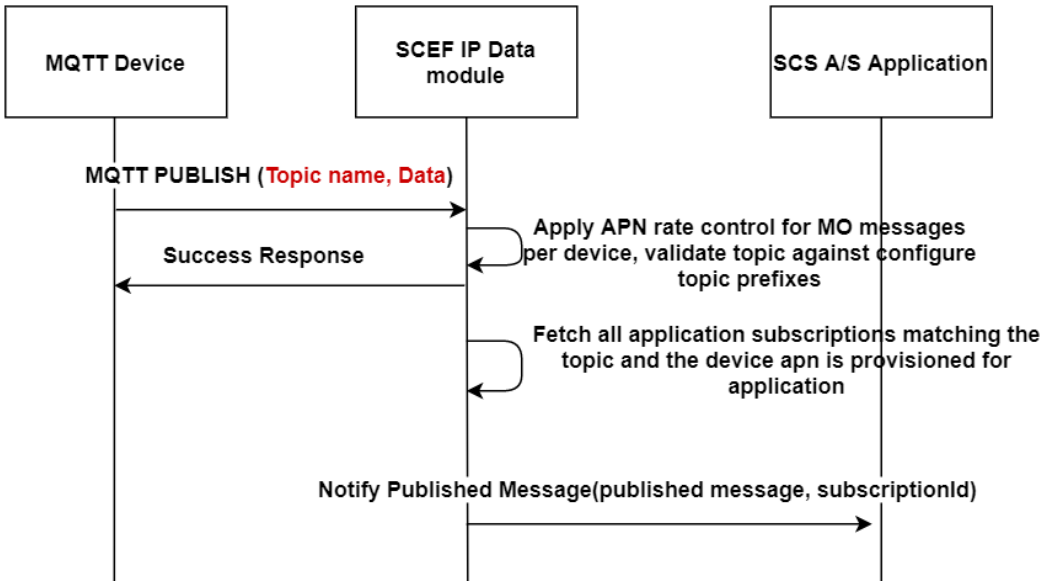
Response Code	Reason
200	Successful
500	Processing error

> **Note:**
>
> When a device reaches the allowed apn rate control limit, messages are not sent for that corresponding device until the completion of configured apn rate control period.

Notifications

The following call flow explains the procedure followed in message notification.

Figure 2-29 Notifications Call Flow



> **Note:**
>
> Subscription with device have more priority than the subscription without device, if it is for the same topic. For information, refer to Figure 2-37

Example 1

Two subscriptions have been created for the same SCSAS for the same External ID. The first subscription is created for the Tracking/Test/# topic and the second subscription is created for the Tracking/Test/Battery/# topic. Then, Device Published data to Tracking/Test/Battery/Level.

The following image displays the entry from table for a subscription created on the Tracking/Test/# topic:

Figure 2-30 Tracking/Test/# for the Same SCSAS

```
| subscription_id                      | callbackurl                                | topic          | appid           | dev
iceid                    |
+--------------------------------------+--------------------------------------------+----------------+-----------------+----
-------------------------+
| 0469641d-7112-433f-add1-35494b46103e | http://200.168.102.8:10001/scs/resources/t8 | Tracking/Test/# | partner1-scefapp | 959
0523908@StreetLight-BLR-4 |
```

The following image displays the entry from table for a subscription created on the Tracking/Test/Battery/# topic:

Figure 2-31 Tracking/Test/Battery/# for the Same SCSAS

```
| subscription_id                      | callbackurl                                | topic                  | appid
   | deviceid                    |
+--------------------------------------+--------------------------------------------+------------------------+--------------
---+-----------------------------+
| b375a2d7-db88-4064-bd6f-175cf4b22b52 | http://200.168.102.8:10001/scs/resources/t8 | Tracking/Test/Battery/# | partner1-scefa
pp | 9590523908@StreetLight-BLR-4 |
```

Notification is sent for subscription created for the Tracking/Test/Battery/# topic when publish message is sent for the Tracking/Test/Battery/Level topic.

Figure 2-32 Tracking/Test/Battery/Level Topic for the Same SCSAS

```
[10-06 02:00:31,527:DEBUG ] Recevied Post request headers [{Content-Type=[application/json], Content-Length=[258], Host=[200.16
8.102.8:10001], Connection=[Keep-Alive], User-Agent=[Apache-HttpClient/4.5.5 (Java/1.8.0_181)], Accept-Encoding=[gzip,deflate]}
] and content [{
  "subscription" : "b375a2d7-db88-4064-bd6f-175cf4b22b52",
  "monitoringEventReports" : [ {
    "externalIds" : [ "9590523908@StreetLight-BLR-4" ],
    "monitoringType" : "Tracking/Test/Battery/Level",
    "data" : "Publish message sent at 11:35pm"
  } ]
}]
```

Example 2

Two subscriptions have been created for two different SCSAS for the same External ID and same topic. Then, Device published data to the topic.

The following image displays the entry from table for a subscription created on the Tracking/Test/Battery/# topic:

Figure 2-33 Tracking/Test/Battery/# Topic

```
| subscription_id                    | callbackurl                                | topic                  | appid
    | deviceid                   | stored_ts  |
+------------------------------------+--------------------------------------------+------------------------+-------------
----+----------------------------+------------+
| 4eafe09e-41f3-4c5f-8feb-d5b33e4edc2f | http://200.168.102.12:10001/scs/resources/t8 | Tracking/Test/Battery/# | partner2-scef
app2 | 7829139111@StreetLight-BLR-6 | 1601966582 |
```

The following image displays the entry from table for a subscription created on the Tracking/Test/Battery/# topic for different SCSAS:

Figure 2-34 Tracking/Test/Battery/# Topic for the Different SCSAS

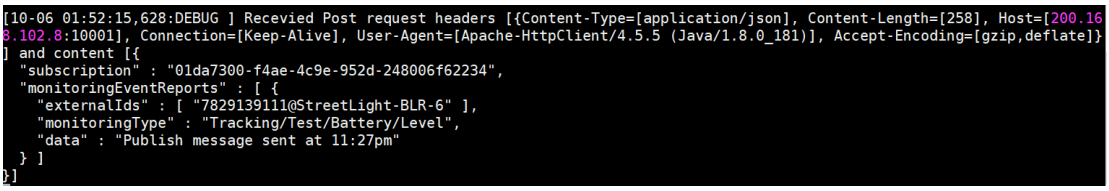
```
| subscription_id                    | callbackurl                               | topic                  | appid
   | deviceid                   |
+------------------------------------+-------------------------------------------+------------------------+--------------
---+----------------------------+
| 01da7300-f4ae-4c9e-952d-248006f62234 | http://200.168.102.8:10001/scs/resources/t8 | Tracking/Test/Battery/# | partner1-scefa
pp | 7829139111@StreetLight-BLR-6 |
```

Notifications are sent for both the subscriptions:

Figure 2-35 Subscription 1

```
[10-06 01:52:11,151:DEBUG ] Recevied Post request headers [{Content-Type=[application/json], Content-Length=[258], Host=[200.16
8.102.12:10001], Connection=[Keep-Alive], User-Agent=[Apache-HttpClient/4.5.5 (Java/1.8.0_181)], Accept-Encoding=[gzip,deflate]
}] and content [{
  "subscription" : "4eafe09e-41f3-4c5f-8feb-d5b33e4edc2f",
  "monitoringEventReports" : [ {
    "externalIds" : [ "7829139111@StreetLight-BLR-6" ],
    "monitoringType" : "Tracking/Test/Battery/Level",
    "data" : "Publish message sent at 11:27pm"
  } ]
}]
```

Figure 2-36 Subscription 2

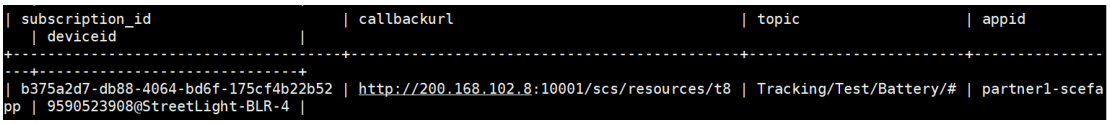


Example 3

Two subscriptions are created. One subscription is with topic for a device, and the other subscription is only for topic.

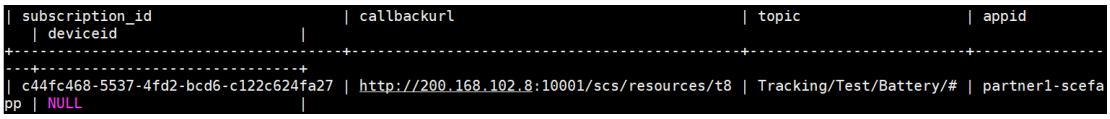
The following image displays the entry from table for a subscription created on the Tracking/Test/Battery/# topic with a device:

Figure 2-37 Tracking/Test/Battery Topic with Device



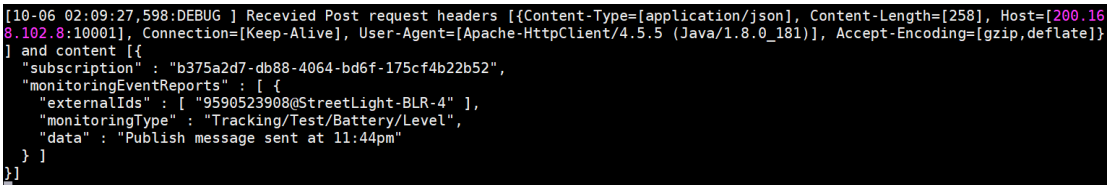
The following image displays the entry from table for a subscription created on the Tracking/Test/Battery/# topic without a device. Ensure that the deviceid is null.

Figure 2-38 Tracking/Test/Battery Topic without Device



The device publishes data to the topic. Notification is sent for the subscription with the device.

Figure 2-39 Subscription Notification



Buffering

For information about buffering of messages, refer to the following sections:

- Data Delivery (Single): Scenario 2 in the Data Delivery (Single) call flow.
- Data Delivery (Broadcast): Last step in the Data Delivery (Broadcast) call flow.

QOS Impact in MQT

The MQTT QoS parameter is applicable for communication between device and mqtt broker. The communication between MQTT broker and SCS A/S is http. Therefore, QoS is not guaranteed for entire path from **Device**, and then **mqtt broker**, and then **application**.
The following table describes different QOS available in MQTT:

Table 2-18 Different QOS available in MQTT

QOS	Description
QoS 0 At most once delivery	The message is delivered according to the capabilities of the underlying network. No response is sent by the receiver and no retry is performed by the sender. The message arrives at the receiver either once or never.
QoS 1 At least once delivery	This quality of service ensures that the message arrives at the receiver at least once. Duplication of messages are possible.
QoS 2 Exactly once delivery	This is the highest quality of service that does not accept loss and duplication of messages. There is an increased overhead associated with this quality of service.

For more information about QOS, refer to MQTT 3.1.1 Specification.

MQTT Features

IP Device Provisioning

Operator need to provision devices in MQTT broker before the device connects to broker.

1. Broker provides a REST based interface for single/batch IP device provisioning/ deletion.
   Refer to the Sample json Body for more details.

   Below device data must be provisioned in broker:

   - Device External Id/MSISDN (either one can be present, if both are present, the `ExternalId` takes precedence)
   - Device Type (MQTT or CoAP) (MQTT-2, CoAP-1)
   - IMSI
   - APN (device that is used to connect to operator's network)
2. Provisioned devices can be updated with details of IMSI, APN and IP address of current device using the PATCH operation for the matching External Id or MSISDN.
   If both External Id and MSISDN are provided, then the External Id is given priority and the devices matching the External Id is updated. Only in the absence of

External Id, the MSISDN is considered. Refer to the Sample json Body for more details.

3. Provisioned devices can be deleted using the POST operation of rest interface for the matching External Id or MSISDN. If both External Id and MSISDN are provided, then External Id is given priority and the devices matching External Id are deleted. In the absence of External Id, MSISDN is considered.

Topic prefixes

Topic prefixes are predetermined prefixes for a given application, which do not contain any wild cards (# or +). The topic names used in subscription and publish message (from application or device) should have a valid prefix.

- The topic prefixes must be configured (mandatory) per application using custom SLA. Topic prefixes must not overlap.
  For example: a/b/c and a/b/c/1 are not allowed.
- When a topic prefix section is deleted from Custom SLA XML then the corresponding topic subscriptions are removed from the DB.
- When SUBSCRIBE/PUBLISH message is received from application, the MQTT broker will check if the topic name in request contains a valid topic prefix and then the topic prefix is configured for the application.
- When SUBSCRIBE/PUBLISH message is received from the device, the MQTT broker checks if the topic name in request contains a valid topic prefix from the configured list of prefixes (across applications). If broker cannot find a valid prefix, the request is rejected.

> **Note:**
>
> Topic prefixes are different from Topic name mentioned for rate control.
> For example:
>
> Topic prefix: `/oracle/india`
>
> Topic name in Rate control section:
>
> `/oracle/india/bangalore/*`

AAA Server Integration

- AAA server integration feature enables identifying MQTT devices with device IP. When device connect to operator network, P-GW (AAA server) will send Diameter ACR Start (Account creation request-271 Command Code) message to AAA interface. The following information from the request will be collected and updated in the device profile.
  - Called-Station-Id (APN)
  - User-Name (IMSI)
  - PDP-Address (IP)
- When MQTT device sends CONNECT to broker, device details are retrieved using the device IP and device is identified with corresponding External Id or MSISDN. External Id or MSISDN is used to communicate MO or MT delivery notifications to SCEF application.
- Diameter ACR Stop - When Diameter ACR Stop (Terminate) message is sent then mapping between IMSI and IP will be removed from the device profile and the device will be disconnected if connected.

Figure 2-40 SCEF MQTT Broker - AAA Server Integration

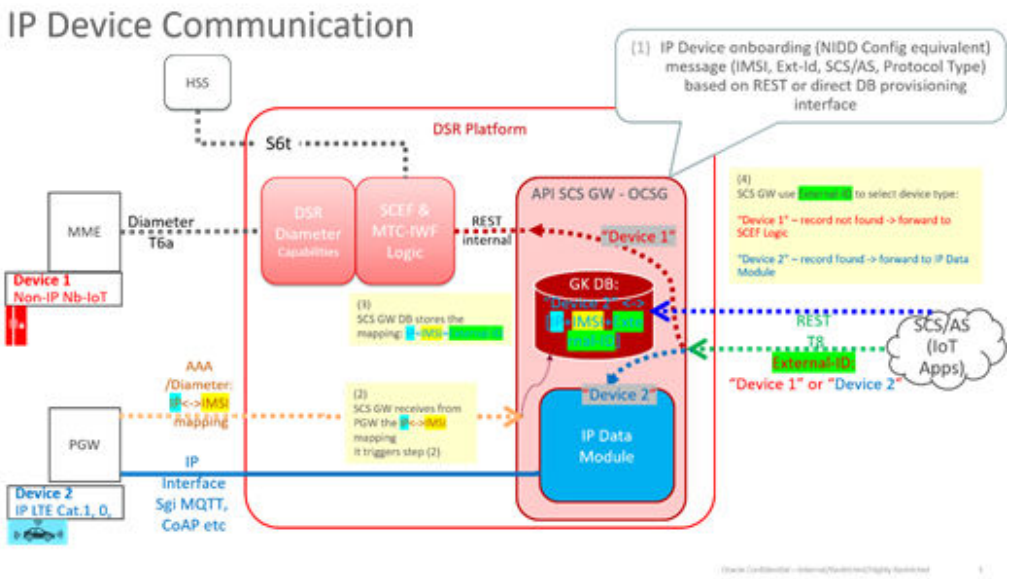
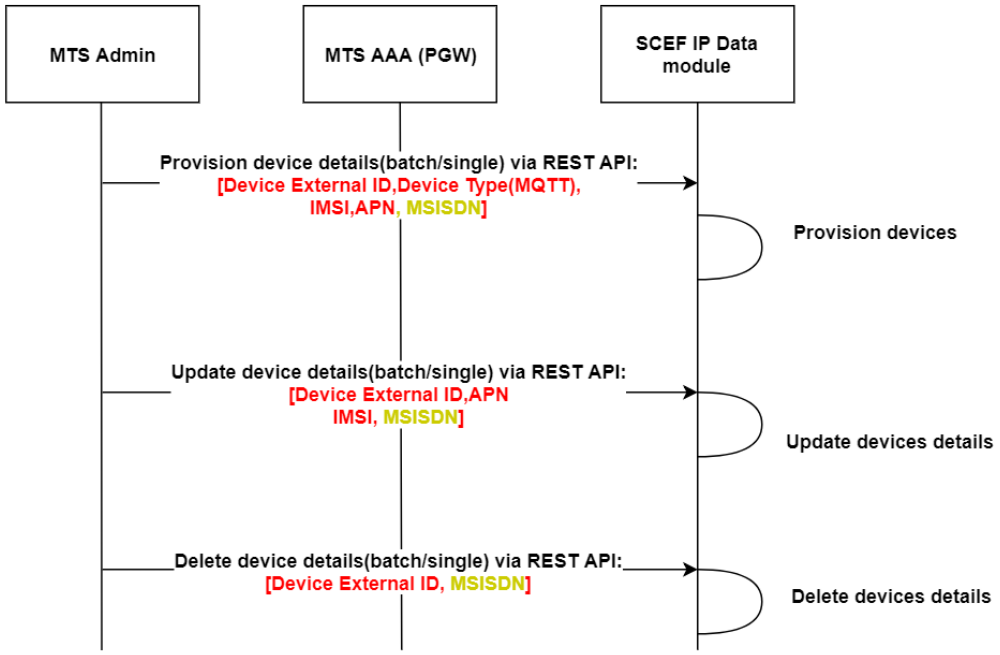


Figure 2-41 SCEF MQTT Broker device provisioning



APN Rate Control, ACL Check for MQTT traffic

APN and ACL checks are applied on the North bound T8 traffic (NIDD Downlink data delivery, ME subscription). These checks are applied only when device id (ExternalId or MSISDN) is present in the T8 the request.

> **Note:**
>
> Device Id is not mandatory in case of NIDD Downlink Data Delivery (DLD) broadcast to all devices subscribed to the topic and ME subscription POST message.

Configuration of APN and ACL per application performed as part of Custom SLA configuration.

APN Check

Input: Configured APN names allowed per application. Device APN is checked against configured APN list per application.

Checkpoint:

1. Delivering MO message from device to application.
2. NIDD DLD message from an application to a device (Single device or broadcast).
3. ME subscription POST to a single device.

Rule: Check if the device APN is allowed for the application.

Result:

1. If APN of device is allowed for the application, proceed further.
2. If APN of device is not allowed for the application, reject or drop the request.

ACL Check

Input: Configured Domain names or MSISDN ranges allowed per application

Checkpoint:

1. Delivering MO message from device to application.
2. NIDD DLD message from an application to a device (Single device or broadcast).
3. ME subscription POST to a single device.

Rule: If message has the device External Id, check the domain name against the configured list per application. If the message has MSISDN then check against the configured MSISDN ranges for the application.

Note: If both the ExternalId and MSISDN are present, then ExternalId is checked and MSISDN is ignored.

Result:

1. If the configured domain names and MSISDN range of application matches the device then proceed further.
2. If the configured domain names and MSISDN range of application does not matches the device then reject or drop the request.

APN Rate control

APN Rate control is applied per device. Rate control is configurable at MQTT MBean GUI at OCSG Admin portal.

APN rate control check is performed for MQTT CONNECT, SUBSCRIBE, PUBLISH to or from devices.

DB Auditor

MQTT broker has the DB auditor functionality enabled to clear old records after configurable period. The tables that needs to be audited are configured in MBean operation.

SCEF Error Handling for AAA Call Flows

Table 2-19 Diameter Error Code

Scenario	Diameter Error Code	Applicable for ACR Type
Based on the existing IMSI of a device if apn is not matched	4010 DIAMETER_END_USER_SERVICE_DENIED	ACR-Start/ACR-Stop/ACR-Interim
When a device is not provisioned.	5030 DIAMETER_USER_UNKNOWN	ACR-Start/ACR-Stop/ACR-Interim
When a session ID is not found.	5002 UNKNOWN_SESSION_ID	ACR-Stop/ACR-Interim
Based on existing IMSI of a device if IP is not matched	4010 DIAMETER_END_USER_SERVICE_DENIED	ACR-Interim

Sample json Body

POST operation of IP device provisioning

```
Rest Interface Uri : https://<APPSERVER IP>:9002/scef/aaa/
deviceprovisioning
This interface is used for addition and deletion of devices.
            {
    "add": {
        "devices": [{
                  "externalId": "1234567891@StreetLight-BLR-1",
                    "msisdn": 1234567891,
                    "imsi": 1234567891,
                    "type": 2,
                    "apn": "test.test.com"
            },
               {
                  "externalId": "1234543210@StreetLight-BLR-2",
                    "imsi": 1234543210,
                    "type": 2,
                    "apn": "test.test1.com"
            }]
    },
    "delete": {
        "ids": [{
                 "externalId": "1234567891@StreetLight-BLR-2",
            },
```

```
            {
                "msisdn": 1987654321
            }]
    }
}
```

PATCH operation of SCEF AAA device provisioning

```
Rest Interface Uri : https://<APPSERVER IP>:9002/scef/aaa/deviceprovisioning
This interface is used for updation of devices data.
{
"update": {
        "devices": [
            {
                "msisdn": 1234543210,
                "imsi": 7829139104,
                "apn": "test.test.com",
                "ip": "10.75.191.1"
            },
            {
                "msisdn": 1234567891,
                "imsi": 1234561234,
                "apn":"test.test1.com"
            }]
    }
}
```

MQTT General Guidelines

General

OCSG Appserver startup time may be extended due to screening of all existing subscriptions from DB.

CustomSLA

1. Topic prefix mandatory in CustomSLA for mqtt feature, without topic prefix in CustomSLA mqtt feature will not function as expected.
2. When mqtt Topic prefix is removed from CustomSLA then corresponding subscriptions will be removed.
3. Subscriptions and PUBLISH data from SCS/AS or Device will be rejected if the topic name does not contain any valid topic prefix.

NIDD/ME for IPDD

1. Subscriptions without device details are treated as topic level subscription. Such that all publish messages from allowed devices for SCS/AS that match to the subscribed topic will be notified to the application.
2. NIDD DLD without device details are treated as broadcast messages. MT messages will be send to devices subscribed to the publishing topic.
3. Custom Header `"Xscef-message-type: MQTT"` is mandatory in case of broadcast MT messages and subscriptions without device details.

4. In the NIDD downlink data deliveries on the request url, config id needs to be provided with value `ipdd`. Refer to Request Mapping section. If the application wants to receive notifications on buffered data delivery, self element needs to be populated with notification url in request body.
5. For NIDD Buffered data delivery status notification, topic name will be provided in the link to indicate topic for which the buffered message is delivered.
6. Buffered messages for devices are deleted after configurable period. Refer to DB Auditor section.

Application Retain Messages

1. NIDD DLD broadcast messages will be persisted in broker.
2. Only latest DLD message is retained at broker per topic.
3. Upon receiving new subscription from device, matching retained messages are delivered to device.
4. To clear retain message, SCS/AS should send DLD broadcast with empty data.
   ```
   { "data": "", "self": "http://192.168.102.5:10001/scs/resources/t8" }
   ```

APN Rate Control

APN rate control check is performed for MQTT CONNECT, SUBSCRIBE, and PUBLISH (to/from) from Devices.

MQTT Protocol Specific

1. MQTT client ID in CONNECT is mandatory.
   There are no specific restrictions, but the size of Client ID cannot be more than 256 characters. Also, it is expected that Client ID is unique for each device.
2. Retain message from device not supported.
3. Qos only between Device and Broker is not applicable to T8 traffic.
4. Topic name should not start with `/`.
5. MQTT PUBLISH message does not support Binary format data.
6. As per the specification, connection to the device drops upon unexpected error in message processing at Broker.
7. It is expected not to give subscription topic same as the mqtt topic prefix. Wildcard is added at the end to receive all messages published for the topic prefix. For example, if mqtt topic prefix is "`/oracle/bangalore/valence/`" then subscription topic should be similar to "`/oracle/bangalore/valence/#`".

MQTT PSK

1. PSK for devices can be configured using MQTT MBean operation. Refer to MQTT configuration section for more details.
2. PSK is encrypted and persisted in SCEF database.

Will message

Device will messages are not persisted at MQTT Broker. Once the device is disconnected from the Broker, will message is removed and notified to all subscribed applications.

CDR

CDR's are created for all T8 Traffic.

AAA Device Provisioning

1. Diameter server supports only TCP transport channel.
2. AAA Diameter Stack starts in server mode.
3. External load balancer required between AAA Server and APIGateway (OCSG).

API Gateway Custom SLA

Functional Summary

SCEF API Gateway supports defining custom SLA per partner application with below details and enforce the rules for the T8 requests:

1. White List IP list - Allowed Client IPs for sending T8 requests to SCEF (multiple IPs can be allowed per application).
2. SCS/AS ID - SCS/AS ID allowed in T8 request to SCEF (only one SCS/AS ID per application is allowed).
3. Event Type rate control for Non IP devices - Rate control based on monitoring event type. Event type should be one of the defined events in T8 specification.
4. Event type rate control for MQTT.

> **Note:**
>
> MQTT rate control is based on the topic. Topics can be provided in SLA for exact matching or wild card matching.

   For example:

   Topic Name: `/oracle/india/bangalore/ptp/valence` - Rate control is applied to the traffic with exact matching topic.

   Topic Name: `/oracle/india/bangalore/*` - Rate control is applied to traffic with topic starting with "`/oracle/india/bangalore/`".

5. APN check for IP devices - APN check on MQTT traffic is applied while communicating with device or on MO notifications.
6. ACL check is based on MSISDN and domain identifier for IP devices - ACL check is applied on traffic with MSISDN or ExternalId. The ExternalID that is present in the request is checked against ACL configuration. In absence of ExternalId, MSISDN is verified.

Note: Each section in the CustomSLA is optional. If the corresponding section is provided then the rule such as Whitelist Ip is applied on T8 traffic.

Configuring Custom SLA

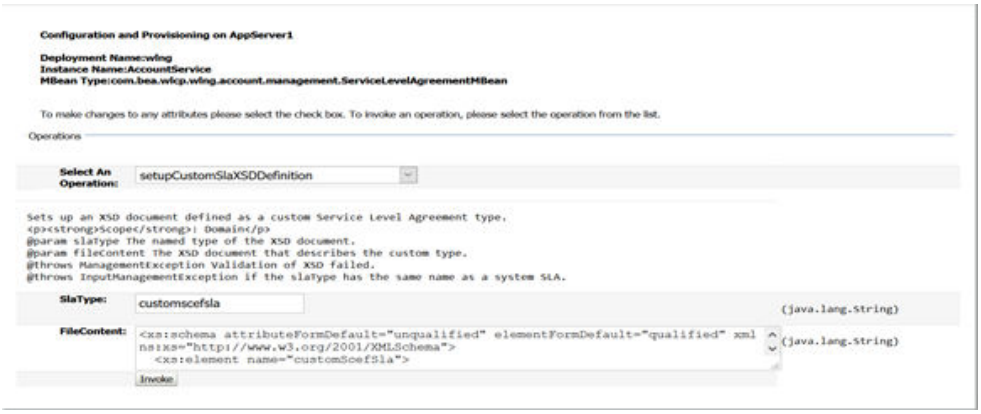
The `SCEFCustomSLAValidation` is a component developed in API Gateway to enforce the required custom rules. The SCEF Custom SLAs addition per application group is configured via admin console.

The following steps are involved:

1. Upload **CustomSLA** XSD:
   a. Login to OCSG Admin Weblogic console.
   b. Navigate to **Domain Structure → OCSG → AppServer1 → Container Services → Account Service → ApplicationSLAs → Operations (Tab)**
   c. Select the operation `setupCustomSlaXSDDefinition` from the list.
   d. Provide information in the following fields as described:
      - **SlaType**: `customscefsl`
      - **FileContent**: Copy the content of XSD file `customslaxsd.xsd` provided in Custom SLA XSD.
   e. After providing the details, click **Invoke**.

   See the below screen for reference.

Figure 2-42 CustomSLA XSD Upload Screen



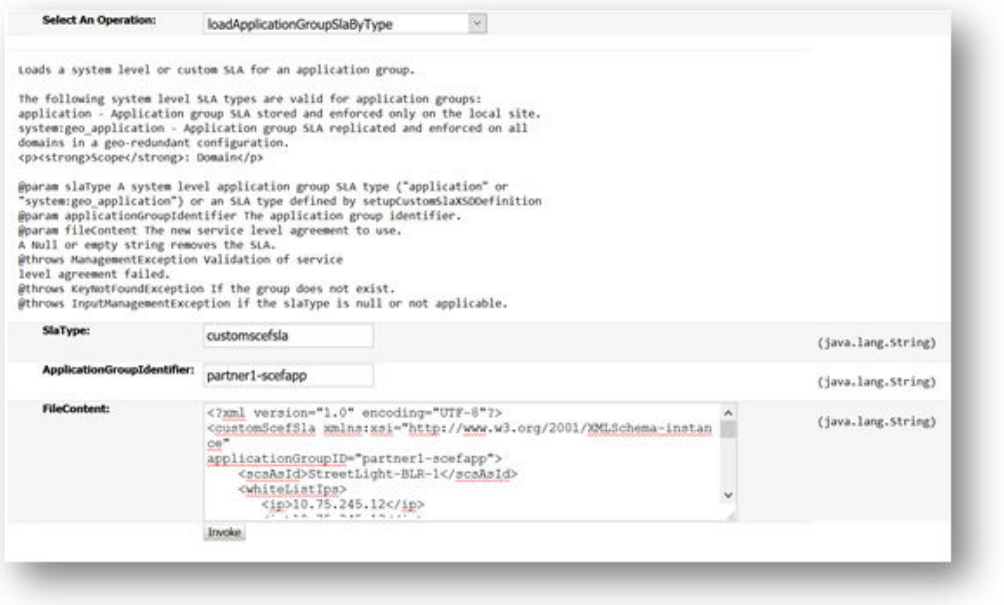
2. Upload **CustomSLA** XML:
   a. Login to OCSG Admin Weblogic console
   b. Navigate to **Domain Structure → OCSG → AppServer1 → Container Services → Account Service → ApplicationSLAs → Operations (Tab)**
   c. Select the operation `loadApplicationGroupSlaByType` from the list
   d. Provide information in the following fields as described:
      - **SlaType**: `customscefsla`
      - **ApplicationGroupIdentifier**: `<partnerusername>-<applicationname>` For example, if `partnerusername` is `partner1` and `applicationname` is `scefapp` then `ApplicationGroupIdentifier` is `partner1-scefapp`

- **FileContent**: Prepare the XML file based on XSD file as provided in Sample Custom SLA XML

e. After providing all the details, click on **Invoke**.

See the below screen for reference.

Figure 2-43 CustomSLA XML Upload Screen



Custom SLA XSD

This section includes the contents of Custom SLA XSD file.

```
<xs:schema attributeFormDefault="unqualified"

elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="customScefSla">
     <xs:complexType>
       <xs:sequence>
         <xs:element type="xs:string" name="scsAsId" maxOccurs="1"
minOccurs="0"/>
         <xs:element name="whiteListIps" maxOccurs="1" minOccurs="0">
           <xs:complexType>
             <xs:sequence>
               <xs:element type="xs:string" name="ip" maxOccurs="unbounded"
minOccurs="0"/>
             </xs:sequence>
           </xs:complexType>
         </xs:element>
         <xs:element name="monitoringRateControl" maxOccurs="1"
minOccurs="0">
           <xs:complexType>
             <xs:sequence>
               <xs:element type="xs:date" name="startDate" maxOccurs="1"
```

```
minOccurs="1" />
               <xs:element type="xs:date" name="endDate" maxOccurs="1"
minOccurs="1"/>
               <xs:element name="monitoringEvents" maxOccurs="1"
minOccurs="1">
                 <xs:complexType>
                   <xs:sequence>
                     <xs:element name="event" maxOccurs="unbounded"
minOccurs="1">
                       <xs:complexType>
                         <xs:sequence>
                           <xs:element type="xs:string" name="name"
maxOccurs="1" minOccurs="1"/>
                           <xs:element type="xs:int" name="rate"
maxOccurs="1" minOccurs="1"/>
                           <xs:element type="xs:long" name="period"
maxOccurs="1" minOccurs="1"/>
                         </xs:sequence>
                       </xs:complexType>
                     </xs:element>
                   </xs:sequence>
                 </xs:complexType>
               </xs:element>
             </xs:sequence>
           </xs:complexType>
         </xs:element>
         <xs:element name="ACL"  maxOccurs="1" minOccurs="0">
           <xs:complexType>
             <xs:sequence>
               <xs:element name="msisdnRanges"  maxOccurs="1"
minOccurs="0">
                 <xs:complexType>
                   <xs:sequence>
                     <xs:element name="msisdn" maxOccurs="unbounded"
minOccurs="1">
                       <xs:complexType>
                         <xs:sequence>
                           <xs:element type="xs:long"
name="startRange" maxOccurs="1" minOccurs="1"/>
                           <xs:element type="xs:long" name="endRange"
maxOccurs="1" minOccurs="1"/>
                         </xs:sequence>
                       </xs:complexType>
                     </xs:element>
                   </xs:sequence>
                 </xs:complexType>
               </xs:element>
               <xs:element name="domainsList"  maxOccurs="1"
minOccurs="0">
                 <xs:complexType>
                   <xs:sequence>
                     <xs:element type="xs:string" name="domain"
maxOccurs="unbounded" minOccurs="1"/>
                   </xs:sequence>
                 </xs:complexType>
```

```
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="APNSet" maxOccurs="1" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="apn" maxOccurs="unbounded"
minOccurs="1"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
          <xs:element name="mqtttopicPrefix" maxOccurs="1" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="prefix"
maxOccurs="unbounded" minOccurs="1"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
         <xs:attribute type="xs:string" name="applicationGroupID"
use="required"/>
     </xs:complexType>
   </xs:element>
</xs:schema>
```

Sample Custom SLA XML

This section contains a sample of Custom SLA XML file.

```
<?xml version="1.0" encoding="UTF-8"?>
<customScefSla xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
applicationGroupID="partner1-scefapp">
    <scsAsId>StreetLight-BLR-1</scsAsId>
    <whiteListIps>
       <ip>10.10.10.6</ip>
       <ip>10.75.245.13</ip>
    </whiteListIps>
    <monitoringRateControl>
       <startDate>2019-02-08</startDate>
       <endDate>2020-05-31</endDate>
       <monitoringEvents>
           <event>
             <name>a/b/d/e/f</name>
             <rate>10</rate>
             <period>1000</period>
          </event>
          <event>
             <name>UE_REACHABILITY</name>
             <rate>10</rate>
             <period>1000</period>
          </event>
          <event>
```

```
                <name>LOCATION_REPORTING</name>
                <rate>10</rate>
                <period>1000</period>
            </event>
            <event>
<name>CHANGE_OF_IMSI_IMEI_ASSOCIATION</name>
                <rate>10</rate>
                <period>1000</period>
            </event>
            <event>
                <name>ROAMING_STATUS</name>
                <rate>1</rate>
                <period>100000</period>
            </event>
            <event>
                <name>COMMUNICATION_FAILURE</name>
                <rate>10</rate>
                <period>1000</period>
            </event>
            <event>
<name>AVAILABILITY_AFTER_DDN_FAILURE</name>
                <rate>10</rate>
                <period>1000</period>
            </event>
            <event>
                <name>NUMBER_OF_UES_IN_AN_AREA</name>
                <rate>10</rate>
                <period>1000</period>
            </event>
        </monitoringEvents>
    </monitoringRateControl>
    <ACL>
     <msisdnRanges>
         <msisdn>
             <startRange>0</startRange>
             <endRange>9500000000</endRange>
         </msisdn>
         <msisdn>
             <startRange>8000000000</startRange>
             <endRange>9000000000</endRange>
         </msisdn>
     </msisdnRanges>
     <domainsList>
         <domain>@oracle</domain>
     </domainsList>
    </ACL>
    <APNSet>
     <apn>apn1@oracle.com</apn>
    </APNSet>
    <mqtttopicPrefix>
         <prefix>a/b</prefix>
         <prefix>a/c/d</prefix>
    </mqtttopicPrefix>
</customScefSla>
```

API-Based Charging

SCEF supports API-based charging for the following events that operate across the T8 reference point:

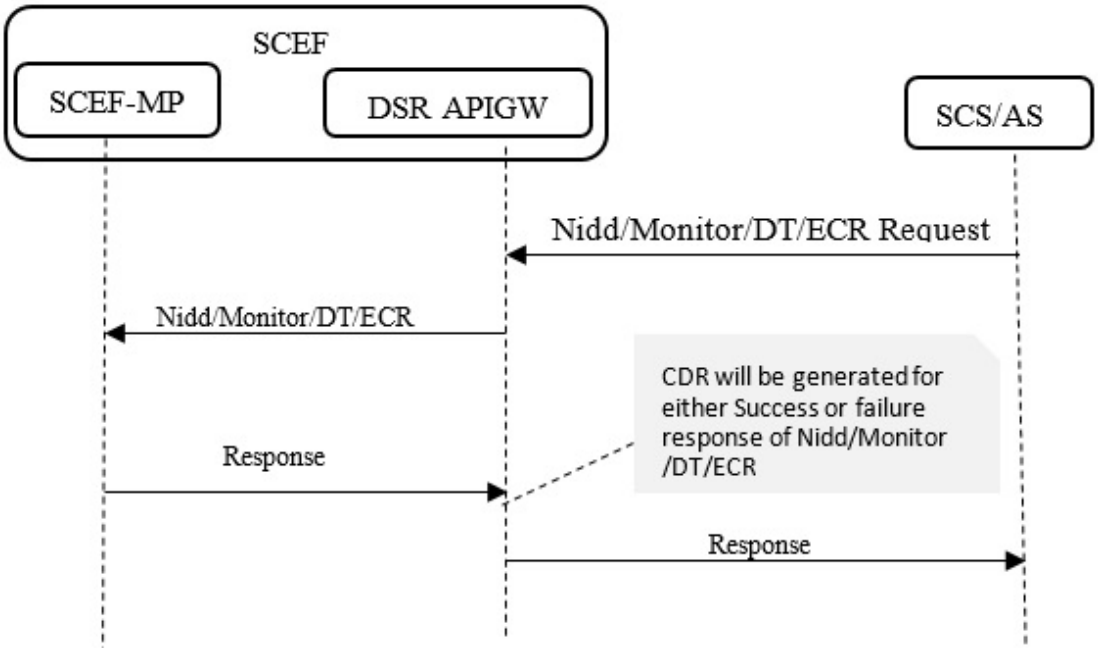
- NIDD Events
- Monitoring Events
- Device Triggering Events
- Enhanced coverage Restriction Events.

In addition, charging is implemented based on offline event-based charging mechanism. The network reports the resource usage by the particular user by forwarding the CDR (Charging Data Record) to the billing domain. Refer to CDR Field Properties for CDR field information.

API-Based Charging for Invocation Events

SCS/AS sends the T8 request (Nidd/Monitor/Device Trigger/ECR) to DSR APIGW, which is part of SCEF. DSR APIGW forwards the incoming T8 request to the serving SCEF-MP. The SCEF-MP acts on the message and sends back a response message to DSR APIGW. DSR then APIGW generates a CDR with available data and writes into a file in binary format. The response is then forwarded to SCS/AS.

Figure 2-44 API-Based Charging for Invocation Events

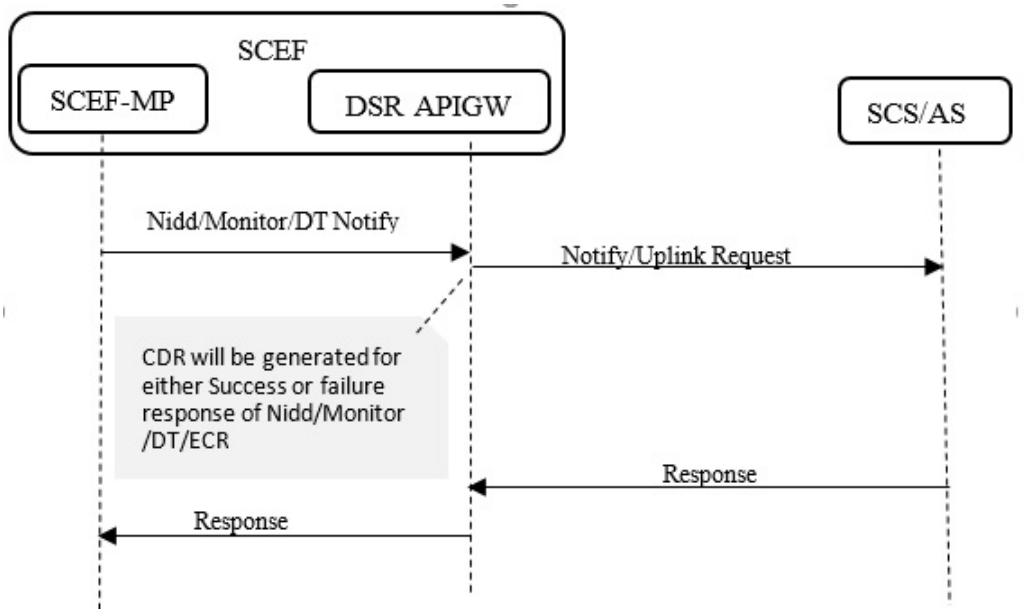


API-Based Charging for Notification Events

SCEF-MP receives a notification message from MME/HSS if SCEF is subscribed for certain events. SCEF-MP sends the notification request (Nidd/Monitor/Device Trigger) to DSR APIGW. DSR APIGW generates a CDR with available data and writes into a file in binary

format. Then request is then forwarded to SCS/AS and SCS/AS response is forwarded to the SCES-MP using DSR APIGW.

Figure 2-45 API-Based Charging for Notification Events



CDR Field Properties

Table 2-20 describes the properties of CDR field.

Table 2-20 CDR Field Properties

Name	Type	Mandatory (M)/ Optional (O)	Tags	Description
recordType	RecordType	M	0 x 80	SCEF exposure function API record.
retransmission	NULL	O	0 x 81	This parameter, when present, indicates information from re-transmitted accounting requests has been used in this CDR.
serviceContextID	ServiceContextID	O	0 x 82	Refers the Self link.
nodeId	NodeID	O	0 x 83	Name of the recording entity.
sCEFID	DiameterIdentity	M	0 x 84	This parameter identifies of the SCEF used for this API invocation.
sCEFAddress	IPAddress	O	0 x A5	This parameter holds the IP address of SCEF.
aPIIdentifier	OCTET STRING	M	0 x 86	The identity of API for each API invocation.
tLTRI	INTEGER	O	0 x 87	The T8 long term transaction reference ID.
tTRI	INTEGER	O	0 x 88	The T8 transaction reference ID.

Table 2-20 (Cont.) CDR Field Properties

Name	Type	Mandatory (M)/ Optional (O)	Tags	Description
sCSASAddress	SCSASAddress	M	0 x A9	The IP address of SCS/AS
eventTimestamp	TimeStamp	M	0 x 8A	The time stamp of the event reported.
aPIInvocationTimestamp	TimeStamp	O	0 x 8B	The time stamp when the API invocation request is submitted to the SCEF from SCS/AS.
aPIDirection	APIDirection	O	0 x 8C	The direction to indicate the API invocation or API notification. Values: invocation (0), notification (1)
aPINetworkServiceNode	APINetworkServiceNode	O	0 x 8D	The identifier of the network element (for example, SGSN, RCAF) that triggers the API notification. Values: MME(0)/HSS(2)
aPIContent	ServUTF8String	O	0 x 8E	The API content (for example, location, monitoring type) used in the T8 transaction for the API invocation request, if available.
aPISize	INTEGER	O	0 x 8F	The size of API payload.
aPIresultCode	INTEGER	O	0 x 90	The result of API Invocation.
externalIdentifier	SubscriptionID	O	0 x B1	The external identifier of the served party associated to the IMSI, MSISDN, or External Group ID, if available.
localRecordSequenceNumber	LocalSequenceNumber	O	0 x 92	Consecutive record number created by this node. The number is allocated sequentially including all CDR types.
recordExtensions	ManagementExtensions	O	0 x B3	A set of network operator/manufacturer specific extensions to the record. Conditioned upon the existence of an extension.

CDR Configuration

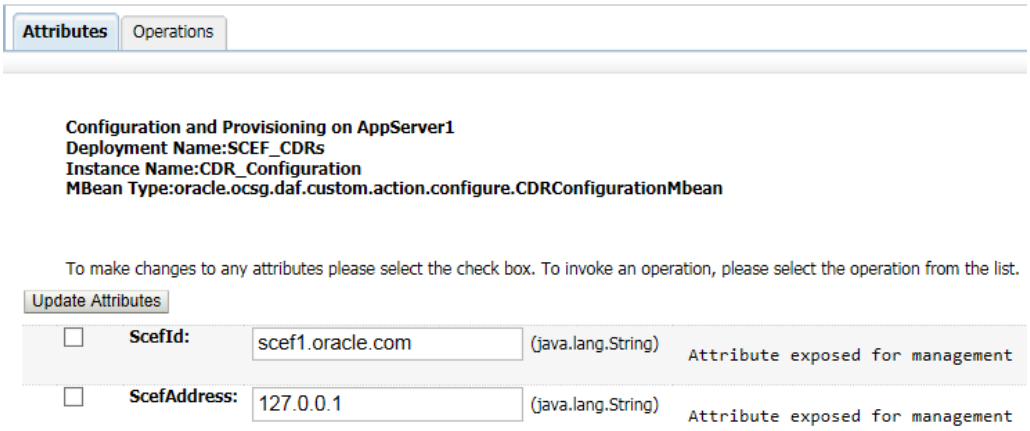
DSR APIGW supports API-based charging for T8 interface messages. Action *SCEFCustomCDR* is a common module developed in the API Gateway to generate API-based CDR for all T8 messages. This module is developed using the API Gateway Actions framework. This action is attached to the T8 API blocks in the API gateway while the Notification request and Invocation response path is set by the API gateway push script.

SCEF has introduced two new fields for CDR configuration that need manual inputs.

- ScefId - Load balancer FQDN
- ScefAddress - Load balancer IP address

These fields are specified on the admin console. The values are updated at runtime and written as CDR after the value is updated.

Figure 2-46 CDR Configuration



CDR Rollout

The CDR file rolls over to an archive folder based on trigger policies. The trigger policies decide when to roll out the file. There are three trigger policies:

1. Line based (NoOfLine)
2. Size based (size in Kb and Mb)
3. Time based

If any of these conditions are met, then the CDR file rolls over to the archive folder. These configurations are in the `log4j2config.xml` file on the appserver. Roll out of the files is decided based on the DefaultRolloverStrategy. The files are stored in the `/u04/{Appserver name}/current/cdr.log` file and are continuously monitored based on the trigger policy. If the file matches with any of the trigger policies, it is archived at `[ <Appserver Name>_-_<RC>.<date>_-_<time> ]`. For example, `AppServer1_-_1.20190110_-_0100-0500`.

CDR Persistent Storage

SCEF uses persistent storage to store the generated CDRs. It is achieved by using an OpenStack Volume.

Every AppServer has its own volume attached to it and every AppServer writes CDRs only on its own attached volume. u04 folder with volume to be created on all App servers.

CDR Transfer with SFTP Tool

The AppServer transfers the generated CDRs to a remote server using SFTP protocol. Bash script `CdrSftpTool.sh`, which is located in appserver, performs the SFTP operations to transfer CDRs at regular intervals.

QoS Control

The QoS Control feature in SCEF allows SCS/AS to set up an AS session with required QoS and priority handling. To achieve this, SCEF acts as an AF to PCRF and sets up the session over an Rx interface. SCEF supports these functionalities:

1. Setup of an AS session
2. Modify/Update of an AS session
3. Deletion of an AS session
4. Event Notification

Table 2-21 Supported T8 Resources and Methods for QoS Control

Resource Name	Resource URI	HTTP Method(s)	HTTP Initiator
AS Session with Required QoS Subscription	3gpp-as-session-with-qos /v1/{scsAsId}/subscriptions	POST	SCS/AS
Individual AS Session with Required QoS Subscription	3gpp-as-session-with-qos /v1/{scsAsId}/subscriptions/{subscriptionId}	PUT, DELETE	SCS/AS
Monitoring Event Notification	{notification_url}	POST	SCEF

Refer to QoS Control Configurationfor Qos configuration details.

AS Session Setup

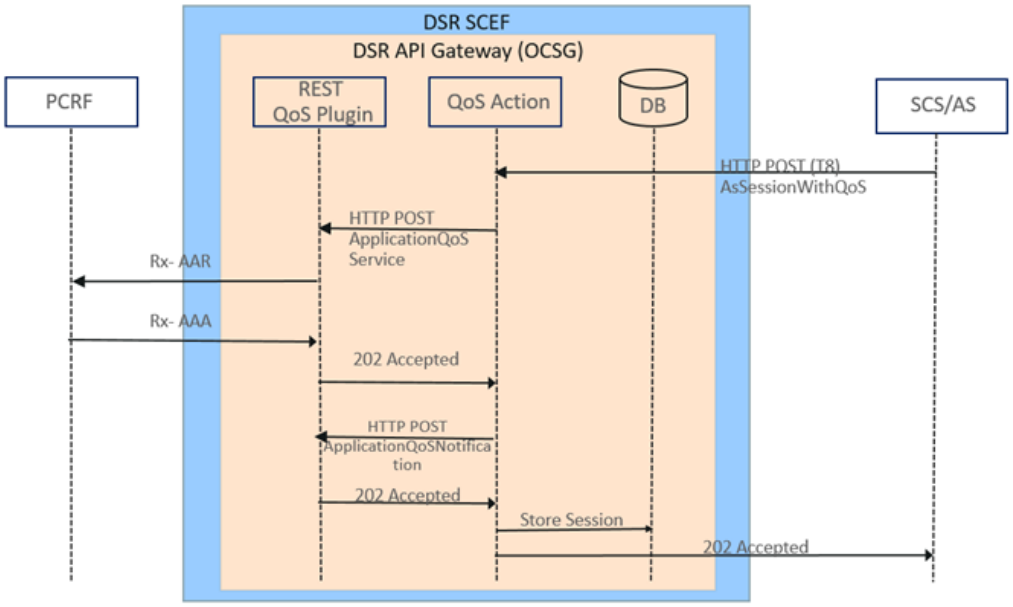
Figure 2-47 illustrates the AS session setup procedure initiated by SCS/AS to a given user as identified using an IPv4 or IPv6 address.

SCS/AS sends the AS Session setup request with required QoS subscription to SCEF. The QoS action implemented at the DSR API Gateway intercepts the T8 request message and converts it to ApplicationQosService REST API to invoke the OCSG's Quality of Service (QoS) communication service. This message contains the QoS properties fetched from the QoS reference configuration based on the QoS reference received in the T8 message or SCS/AS configuration. The QoS plugin from the DSR API Gateway (OCSG) generates an Rx-AAR message toward PCRF to create an AS session with the requested QoS and priority signaling.

When PCRF sends a success response in Rx-AAA, the QoS plugin sends a success response to the ApplicationQoSService request. The QoS action then creates an ApplicationQoSNotification to request for notifications in the event of a bearer release, session expiry. A session record is then created in the database and a resource location (URI) is sent to SCS/AS in the response.

The SCS/AS uses the URI received from the Location header in subsequent requests to the SCEF to refer to this AS session. Otherwise, the SCEF sends an HTTP response to the SCS/AS with a corresponding status code and includes the result in the body of the HTTP response.

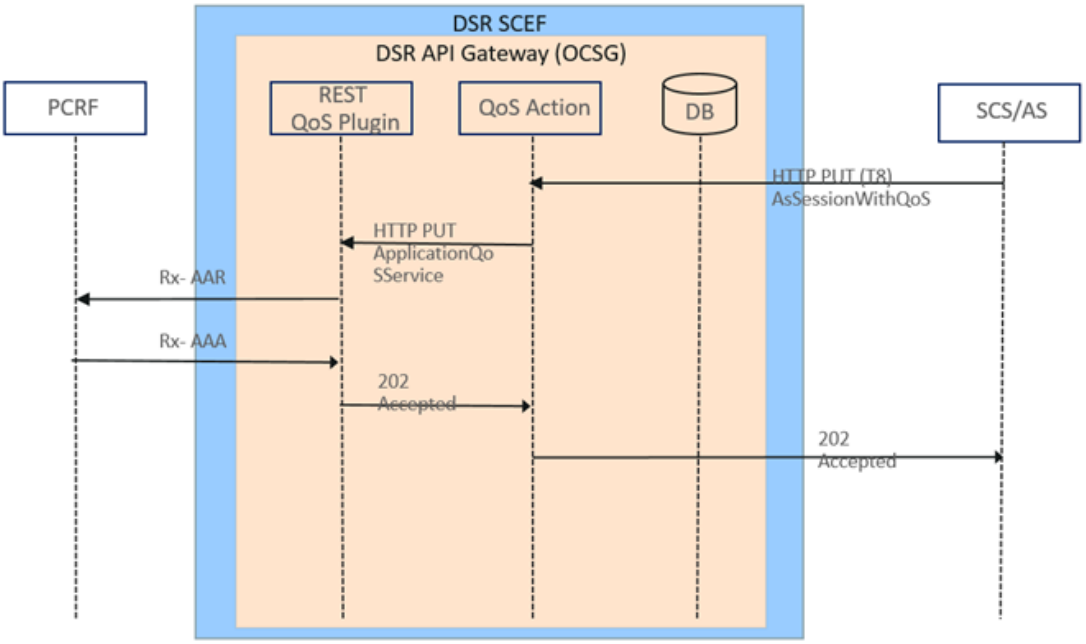
Figure 2-47 AS Session Setup



AS Session Modify

Figure 2-48 illustrates the AS Session modify procedure to replace existing QoS properties. To update the established AS session, the SCS/AS sends an HTTP PUT message to the *Individual AS Session with Required QoS Subscription* resource requesting to replace all properties in the existing resource, addressed by the URI received in the response to the request that has created the resource. The UE IP address has to remain unchanged from previously provided values. After receiving such a message, the QoS Action in OCSG converts it to a HTTP PUT message to an *Application QoS Service* resource and sends it to the REST QoS plugin. This message contains the QoS properties fetched from the QoS reference configuration based on the QoS reference received in T8 message or SCS/AS configuration. The REST QoS plugin then makes the change and interacts with the PCRF to modify the Rx session by triggering an Rx-AAR.

Figure 2-48 AS Session Modify

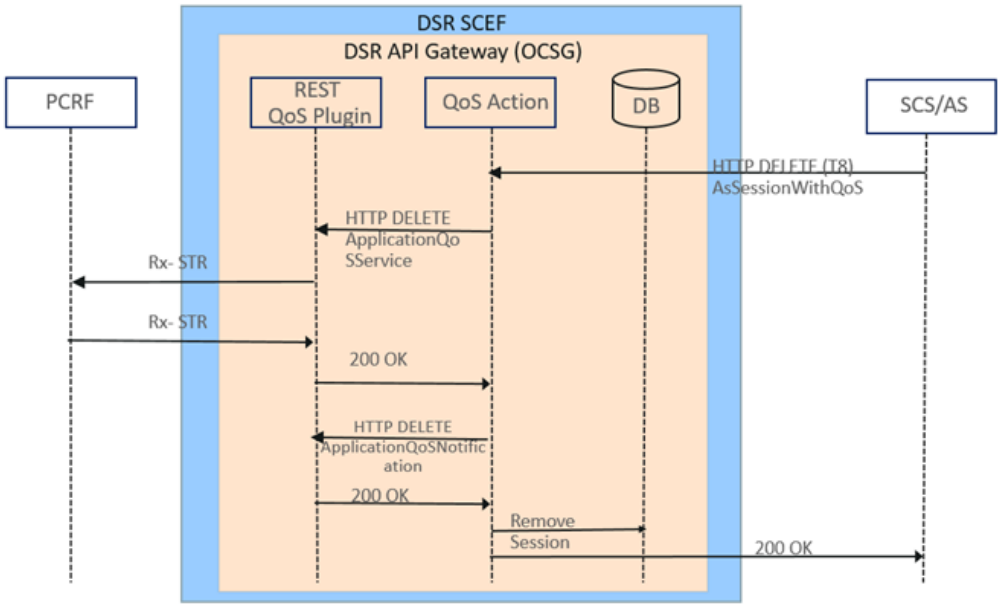


AS Session Remove

Figure 2-49 illustrates the AS Session Remove procedure initiated by SCS/AS. To remove the AS session, SCS/AS sends an HTTP DELETE message to the *Individual AS Session with Required QoS Subscription* resource. After receiving the HTTP DELETE message, the SCEF QoS action sends an HTTP DELETE for the *Application QoS Service* resource toward the REST QoS plugin. The REST QoS plugin then removes all properties and interacts with the PCRF to terminate the Rx session by sending an Rx-STR command.

After a successful response is received from the REST QoS plugin, an HTTP DELETE for the *ApplicationQoSNotification* resource is sent by the QoS action to unregister notifications for the UE and the AS session data stored in database is cleared. A success response is then sent to the SCS/AS stating that the AS session has been removed successfully.

Figure 2-49 AS Session Remove

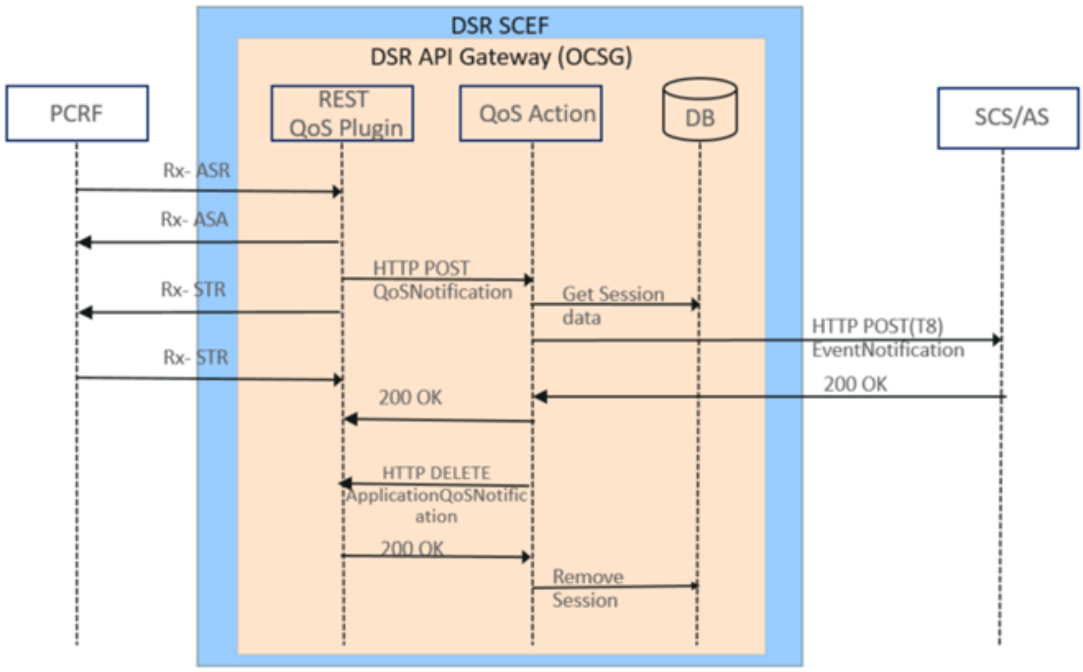


AS Session QoS Modification

When the REST QoS plugin receives a traffic plane notification (for example, transmission resource lost), or if it gets informed that the Rx session is terminated (for example, due to a release of PDN connection), then it sends an HTTP POST message including the notified event (for example, session terminated) and the accumulated usage (if received from the PCRF) to the callback URI *notificationUri* provided by the SCS/AS during the creation of the individual AS Session with the Required QoS Subscription. The SCS/AS responds with an HTTP response to confirm the received notification.

Figure 2-50 illustrates the event notification generated when a PCRF terminates the Rx session. In this case, PCRF sends a Rx-ASR command to the REST QoS service plugin. The QoS service plugin then triggers a notification toward the QoS action, which verifies if the session exists in the database. Once the session data is fetched from the database, a notification is sent to the SCS/AS with details of the event. After a response is received from the SCS/AS, an unregister notification request is sent to the REST QoS plugin to stop sending notifications for this UE. The session data in the database is then cleared.

Figure 2-50 AS Session QoS Modification



3
Managed Objects

SCEF works with Common (including SCS/AS and System Options), AppWorks, and NIDD (including NIDD and APN Configuration) managed objects described in this chapter.

SCS/AS

The SCS/AS managed objects exist for each SCS/AS that needs to communicate with DSR's SCEF application. This managed object allows the customer to configure an SCS/AS by specifying its properties and associate an APN to it. Attributes listed in SCS/AS are used to configure the SCS/AS managed object.

Table 3-1 SCS/AS Attribute Descriptions

Attribute	Description
scsAsId	The SCS/AS identifier.
niddCfgSetName	The NIDD Configuration Set managed object associated to this SCS/AS. When this attribute is populated, the NIDD feature is enabled.
apnCfgSetName	The APN Configuration Set managed object associated to this SCS/AS. This attribute must be populated if the niddConfigSetName is populated.
monitoringEventCfgSetName	The Monitoring Events Configuration Set managed object associated to this SCS/AS.
deviceTriggeringCfgSetName	The Device Triggering Configuration Set managed object associated to this SCS/AS.
aclId	Associated AclId with SCS.
callbackUrl	Destination URL for any notification messages for this SCS/AS.
smsScFqdn	FQDN of SMS-SC.
smsScRealm	Realm of SMS-SC.
scsAsIsdn	ISDN number of the SCS/AS in international ISDN number format.
isEcrAllowed	Value of this attribute decides if Enhanced Coverage Restriction Control is allowed or not.

System Options

The System Options managed objects allow the customer to specify routing configurations and system defaults that apply to a DSR node. Attributes listed Table 3-2in are used to configure the System Options managed object.

Table 3-2 System Options Attribute Descriptions

Attribute	Description
art	Application Routing Table instance used to route any Diameter request messages generated by the SCEF application.
prt	Peer Routing Table instance used to route any Diameter request messages generated by the SCEF application.
apiGwIpList	A comma separated list of IPv4 addresses of the DSR API Gateway application server. The SCEF application distributes the HTTP request messages toward the DSR API Gateway among the IP addresses listed in the *apiGwIpList* attribute.
priority	DRMP priority of NIR, ACR, and CMR messages originated by SCEF.
retryDbUpdate	The number of times the SCEF MP server may retry when an attempt to update a context in the USBR server fails due to concurrent update checksum mismatch.
servingPlmnRateControlEnabled	This option allows the customer to enable or disable Mobile Terminating PDU rate control based on the Serving PLMN Rate Control configuration requested by MME/SGSN.
scefWaitTime	The duration of time in seconds the SCEF application may wait for a Diameter Answer message for any request sent to the MME/SGSN. The *scefWaitTime* attribute should match the Pending Answer Timeout value in the Diameter configuration.
binaryEncoder	The Binary-To-Text encoding scheme to use to transcode binary data while sending to or receiving from the SCS/AS in JSON-encoded HTTP message. Allowed values are: • Base2 • Base16 • Base64 • ASCII

AppWorks

The AppWorks managed objects (Server Groups, Resource Domains, Places, and Place Associations) require the following changes described in this section.

- All DA-MP servers that running the SCEF application need to be configured in a Place Association of type *Application Region*.
- All DA-MP servers that running the SCEF application need to be configured in a Resource Domain with a *Application MPs* profile.

NIDD Configuration Set

An NIDD Configuration Set managed object exists for each instance of an NIDD Configuration Set. This managed object allows the customer to create instances of NIDD Configuration Sets as needed. The NIDD Configuration Set specifies various

attributes that controls the processing of NIDD-related messages, and safeguards the operator from unacceptable request parameters, for example, an authorization duration that is too long.

Table 3-3 Non-IP Data Delivery Attribute Descriptions

Attribute	Description
name	A name that uniquely defines the NIDD Configuration Set.
maxAuthDuration	Maximum time in seconds the NIDD configuration is valid.
pdnEstablishmentOptionEnabled	Applicability of the PDN Establishment Option IE received in the NIDD Configuration message and/or Downlink Data Delivery messages received for this NIDD configuration.
pdnEstablishmentOption	Default PDN Establishment Option to apply if the NIDD Configuration message and any subsequent Downlink Data Delivery message(s) do not contain the PDN Establishment Option IE. **Note**: This attribute is applicable only if the *pdnEstablishmentOptionEnabled* is set to true. Any change made to the *pdnEstablishmentOptionEnabled* attribute takes effect when the next Downlink Data message is received from the SCS/AS.
dataDuration	The maximum time in seconds a Downlink Data Delivery message is considered to be valid when it is buffered by the SCEF application. If *dataDuration* is set to 0, then SCEF does not buffer NIDD MT data messages irrespective of *maxOnholdDataMsg* configuration and maximum message latency value received in the MT NIDD submit request from SCS/AS.
maxOnholdDataMsg	The maximum number of messages for each UE that can be buffered by SCEF application.
priority	The default priority associated to a Downlink Data Delivery message when the same is not present in the message received from the SCS/AS.
minRetransmissionTime	The minimum time in seconds the SCEF application requires to buffer a Downlink Data Delivery message in the USBR database and then retransmit it to the MME/SGSN. **Note**: This attribute becomes more significant in slow WAN networks.

APN Configuration

An APN Configuration Set managed object instance exists for each Access Point Name that the customer servers by SCEF signaling. An APN Configuration Set managed object is associated to an SCS/AS. This managed object specifies various attributes that controls the processing of signaling messages that belong to that APN, for example the rate control attributes.

Table 3-4 Access Point Name Attribute Descriptions

Attribute	Description
Name	Access Point Name.
maxPacketSize	Maximum Packet Size (Uplink or Downlink) in bytes that is allowed to be transmitted through the SCEF application.
maxPacketBufferSize	Maximum Packet Size (Downlink) in bytes that is allowed to be buffered by the SCEF application.
downlinkApnRateControlUnit	The unit for Downlink APN rate control. Any change made to the *downlinkApnRateControlUnit* attribute takes effect when the next Downlink Data message is received from the SCS/AS.
downlinkApnRateControlVal	Multiple of Downlink APN rate control unit. Any change made to the *downlinkApnRateControlVal* attribute takes effect when the next Downlink Data message is received from the SCS/AS.
downlinkApnMessageRate	The maximum Downlink message rate for this APN. Any change made to the *downlinkApnMessageRate* attribute takes effect when the next Downlink Data message is received from the SCS/AS.
uplinkApnRateControlUnit	The unit for Uplink APN rate control. Any change made to the *uplinkApnRateControlUnit* attribute takes effect when the next Downlink Data message is received from the SCS/AS.
uplinkApnRateControlVal	Multiple of Uplink APN rate control unit. Any change made to the *uplinkApnRateControlVal* attribute takes effect when the next Downlink Data message is received from the SCS/AS.
uplinkApnMessageRate	The maximum Uplink message rate for this APN. Any change made to the *uplinkApnMessageRate* attribute takes effect when the next Downlink Data message is received from the SCS/AS.

4
Configure SCEF

All configurations and status reporting for the SCEF application is performed using machine-to-machine interfaces. To access the MMI API documentation through a direct URL access, without login, go to http:///(IP address of NOAM or SOAM)/raml/mmi.html. Or the MMI API documentation can be accessed directly from the DSR GUI by clicking on the new MMI API Guide menu item.

The SCEF application is bundled with DSR. Once you have activated SCEF and start using it, SCEF changes, described in this section, are seen in the DSR GUI.

Alarms, Events, and KPI Changes

Alarms, Events, and KPIs have been added to the Alarms and KPI Reference Manual available on OHC at https://docs.oracle.com/en/industries/communications/diameter-signaling-router/index.html.

Measurement Changes

Measurements have been added to the Measurements Reference Manual available on OHC at https://docs.oracle.com/en/industries/communications/diameter-signaling-router/index.html.

MMI Changes

MMI changes are described in the SCEF MMI Attributes chapter of this manual.

Basic SCEF Configuration

Follow these steps first before beginning the SCEF MMI configuration.

1. Configure the Resource, Resource Domain, Place, and Place Association for SCEF. Refer to the Session Binding Repository (SBR) User's Guide on OHC.
2. Configure the Local Node, Peer Nodes, Connections, PRT, and ART (table and rules). Refer to the Diameter User's Guide on OHC.
3. (Optional) If Application Chaining is intended for Diameter routing:
    a. Configure ART (Table and Rules).
    Refer to the Diameter User's Guide on OHC.
    b. Configure RBAR/FABR.
    Refer to the Range Based Address Resolution User's Guide and Full Address Based Resolution User's Guide on OHC.

SCEF MMI Configuration

Once the basic configuration is complete, make these changes in MMI on the SO. The MMI API documentation can be found on OHC.

1. Configure ACL by configuring these managed objects:
    a. Access Control List

An Access Control List (ACL) configuration entry consists of a name. The purpose of the ACL is to maintain a set of Access Control Rules that can be associated to one or more SCS/AS.

```
Send a POST message to an active SOAM with a URL  scef/
accesscontrollists  and content as specified in
accesscontrollist.json
```

For example,

```
scef/accesscontrollists -v POST -r accesscontrollist.json
```

b. Access Control Rules
This MO maintains all rules for access control.

```
Send a POST message to an active SOAM with a URL  scef/
accesscontrolrules  and content as specified in
accesscontrolrule_DN.json
Send a POST message to an active SOAM with a URL  scef/
accesscontrolrules  and content as specified in
accesscontrolrule_Domain.json
```

For example,

```
scef/accesscontrolrules -v POST -r accesscontrolrule_DN.json
scef/accesscontrolrules -v POST -r accesscontrolrule_Domain.json
```

c. Access Control Associations
This MO creates/maintains the association between the List and Rules. This is used to tell which rules are under a specific List.

```
Send a POST message to an active SOAM with a URL  scef/
accesscontrolassociations  and content as specified in
accesscontrolassociation.json
```

For example,

```
scef/accesscontrolassociations -v POST -r
accesscontrolassociation.json
```

2. Configure APN with the name and configuration-specific data for APN.

```
scef/apnconfigurationsets -v POST -r apnconfigurationset.json
```

3. Configure the Options managed object with system configuration parameters for SCEF.

```
scef/options -v POST -r options.json
```

4. Configure the NIDD configuration set.

```
scef/niddconfigurationsets -v POST -r niddconfigurationset.json
```

5. Configure the Monitoringevent configuration set.

```
scef/monitoringeventconfigurationsets -v POST -r
monitoringeventconfigurationset.json
```

6. Configure the Device Triggering configuration set.

```
scef/devicetriggeringconfigurationsets -v POST -r
devicetriggeringconfigurationset.json
```

7. Configure SCS/AS to specify the SCSid and its associated configuration. Use JSON based on the features that need to be enabled.

```
/scef/scsapplicationservers -v POST -r  scsapplicationserver.json
```

8. Execute the following command to configure the Monitoring Area Pool configuration:

```
scef/monitoringareapools -v POST -r monitoringareapools.json
```

9. Execute the following command to configure the Monitoring Location Area configuration:

```
scef/monitoringlocationareas -v POST -r monitoringlocationareas.json
```

> **Note:**
>
> When a DSR with SCEF application enabled is upgraded from any earlier version to DSR 8.5 or later, then the existing MMI configuration records for `/scef/monitoringlocationareas/` must be deleted before the upgrade. Then, reconfigure the records for `/scef/monitoringlocationareas/` according to the new schema after the upgrade is complete on all the servers.

10. Create JSON files MMI will use.

Example 4-1 scsapplicationserver.json

scsapplicationserver.json with NIDD enabled.

```
{
   "apnCfgSetName": "apn1.test.com",
   "callbackUrl": "https://test.xyz.com",
   "interimInterval": 600,
   "niddCfgSetName": "NIDDCFG1",
   "scsAsId": "SCSAS1"
}
```

scsapplicationserver.json with MONITORING enabled.

```
{
   "apnCfgSetName": "apn1.test.com",
   "callbackUrl": "https://test.xyz.com",
   "interimInterval": 600,
   "monitoringEventCfgSetName": "MONEVENT1",
```

```
   "scsAsId": "SCSAS1"
}
```

scsapplicationserver.json with ECR enabled.

```
{
     "callbackUrl": "https://test.xyz.com",
   "isEcrAllowed":true,
   "scsAsId": "SCSAS1"
}
```

scsapplicationserver.json with DT enabled.

```
{
   "apnCfgSetName": "apn1.test.com",
   "callbackUrl": "https://test.xyz.com",
   "interimInterval": 600,
   "deviceTriggeringCfgSetName": "DevTrig1",
   "scsAsId": "SCSAS1",
   "scsAsIsdn":"123456789",
   "smsScFqdn": "test.one",
   "smsScRealm":"oracle.com"
}
```

scsapplicationserver.json with ACL, NIDD, MON, and DT enabled.

```
{
   "apnCfgSetName": "apn1.test.com",
   "callbackUrl": "https://test.xyz.com",
   "aclName":"ACL1",
   "interimInterval": 600,
   "niddCfgSetName": "NIDDCFG1",
   "monitoringEventCfgSetName": "MONEVENT1",
   "deviceTriggeringCfgSetName": "DevTrig1",
   "scsAsId": "SCSAS1",
   "scsAsIsdn":"123456789",
   "smsScFqdn": "test.one",
   "smsScRealm":"oracle.com"
}
```

Example 4-2 accesscontrolrule_DN.json

This creates a rule to allow NIDD and Monitoring for DOMAIN = test.oracle.com.

```
{
   "name": "RULE2",
   "domain": "test.oracle.com",
   "supportedFeatures": [
          "Monitoring",
          "Nidd"
   ],
```

```
   "userIdentifierType": "DOMAIN"
}
```

Example 4-3 accesscontrolassociation.json

Associates RULE1 with ACL1

```
{
   "aclName": "ACL1",
   "ruleName": "RULE1"
}
```

Example 4-4 apnconfigurationset.json

This managed object controls the APN level rate control parameters.

```
{
   "downlinkApnMessageRate": 300,
   "downlinkApnRateControlUnit": "Day",
   "downlinkApnRateControlVal": 3,
   "maxPacketBufferSize": 100,
   "maxPacketSize": 100,
   "name": "apn1.test.com",
   "value": "apn1.test.com",
   "uplinkApnMessageRate": 600,
   "uplinkApnRateControlUnit": "Week",
   "uplinkApnRateControlVal": 6
}
```

Example 4-5 options.json

```
{
   "apiGwIpList": [
       "20.20.20.2",
       "20.20.20.4"
   ],
   "art": "ART1",
   "prt": "PRT1",
   "scefId": "oracle.com",
   "scefWaitTime": 1200,
   "servingPlmnRateControlEnabled": true
}
```

Example 4-6 monitoringeventconfigurationset.json

```
{
   "monitoringType": [
          "UEReachability",
          "LocationReporting"
   ],
   "name": "MONEVENT1"
}
```

Example 4-7 devicetriggeringconfigurationset.json

```
{
   "defaultApplicationPort": 1000,
   "defaultPriority": "NonPriority",
   "maxValidityPeriod": 3600,
   "name": "DevTrig1",
   "mandateApplicationPort": false,
   "mandatePriority": false
}
```

Example 4-8 niddconfigurationset.json

```
{
   "dataDuration": 20,
   "maxAuthDuration": 86400,
   "maxOnholdDataMsg": 1,
   "minRetransmissionTime": 6,
   "name": "NIDDCFG1",
   "pdnEstablishmentOption": "INDICATE_ERROR",
   "pdnEstablishmentOptionEnabled": false
}
```

Example 4-9 monitoringareapools.json

```
{

"poolName" : "south",
"locationAreaFqdn" : "mmesouth1.oracle.com",
"locationAreaRealm" : "oracle.com"

}
```

Example 4-10 monitoringlocationareas.json

monitoringlocationareas.json with TRACKING AREA ID information

```
{

"locationAreaType": "TRACKINGAREAID",
"mcc": 987,
"mnc": 654,
"poolName": "south",
"trackingAreaCodeEnd": 3060,
"trackingAreaCodeStart": 3050

}
```

monitoringlocationareas.json with ROUTING AREA ID information

```
{

"poolName" : "south",
```

```
"locationAreaType" : "ROUTINGAREAID",
"mcc" : 123,
"mnc" : 125,
"locationAreaCode" : 1500,
"routingAreaCodeStart" : 240,
"routingAreaCodeEnd" : 245

}
```

5
Monitoring Event

The Monitoring Events feature monitors specific events in the 3GPP system and makes monitoring event information available using SCEF. It identifies the 3GPP network element suitable for configuring specific events, event detection, and event reporting to the authorized users, for example, for use by applications or logging. If such an event is detected, the network can be configured to perform special actions, for example, limit UE access.

The Monitoring Event procedure requested by SCS/AS is determined from the URI as described in Table 5-1

Table 5-1 Supported Monitoring Event Resources and Methods

Resource Name	Resource URI	HTTP Method(s)	HTTP Initiator
Monitoring Event Subscriptions	3gpp-monitoring-event/v1/ {scsAsId}/ subscriptions/	POST	SCS/AS
Individual Monitoring Event Subscription	3gpp-monitoring-event/v1/ {scsAsId}/ subscriptions/ {subscriptionId}	GET, DELETE	SCS/AS
Monitoring Event Notification	{notificationDestination}	POST	SCEF

Supported Monitoring Events include:

- LOSS_OF_CONNECTIVITY
- UE_REACHABILITY
- LOCATION_REPORTING
- CHANGE_OF_IMSI_IMEI_ASSOCIATION
- ROAMING_STATUS
- UE_REACHABILITY plus idleStatusIndication flag = true
- COMMUNICATION_FAILURE
- AVAILABILITY_AFTER_DDN_FAILURE plus idleStatusIndication flag = true
- NUMBER_OF_UES_PRESENT_IN_AN_AREA

Monitoring Event Subscription

To subscribe a new monitoring event configuration, the SCS/AS sends an HTTP POST message to the SCEF. The body of the HTTP POST message includes the Monitoring Type, and may include External Identifier(s) or MSISDN(s) or External Group ID, Maximum Number of Reports, Monitoring Duration, T8 Destination Address, and Group Reporting Guard Time, where the External Identifier or MSISDN indicates the subscription for an individual UE and the External Group ID indicates a group of UEs. SCEF generates a corresponding subscription ID for a new subscription request.

> **Note:**
>
> SCEF always gives higher preference to an External Identifier when both Identifiers (External Identifier and MSISDN) are present in the Monitoring Event Configuration Request message.

The SCS/AS sends a Monitoring Subscription Request (External Identifier or MSISDN or External Group ID, Monitoring Type, Maximum Number of Reports, Monitoring Duration, T8 Destination Address, and Group Reporting Guard Time) message to the SCEF.

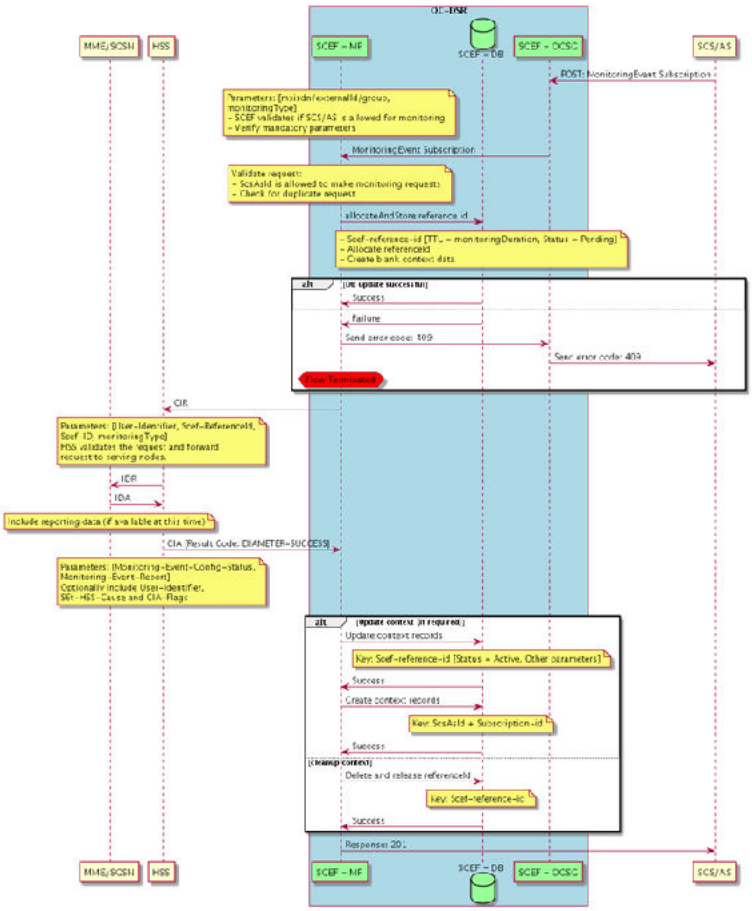
If the SCS/AS wants to configure Monitoring Event for the group of UEs, the SCS/AS can send a Monitoring Request message including External Group Identifier and Group Reporting Guard Time. A Group Reporting Guard Time is an optional parameter to indicate aggregated Monitoring Event Reporting(s), which has been detected for the UEs in a group, needs to be sent to the SCS/AS once the Group Reporting Guard Time is expired.

The SCEF stores the SCS/AS Identifier, T8 Destination Address, Monitoring Duration, and Maximum Number of Reports. The SCEF generates a subscription ID in case of a new POST request.

The SCEF sends a Monitoring Request (External Identifier or MSISDN or External Group Identifier, SCEF ID, SCEF Reference ID, Monitoring Type, Maximum Number of Reports, and Monitoring Duration) message to the HSS to configure the given Monitoring Event on the HSS in Configuration-Information-Request (CIR) message.

After processing, HSS sends a Configuration-Information-Answer (CIA) message. Then according to the result code received in the CIA message, if the result code is Success (2001), the SCEF sends a Monitoring Response (Subscription, Configuration Results, Monitoring Event Reports and Cancel Indication) message to the SCS/AS to acknowledge acceptance of the Monitoring Request; if the result code is not successful, then an error result code informs the SCS/AS about the error occurred/ received.

Figure 5-1 Monitoring Event Subscription



Monitoring Event Notification

Notification in Reporting-Information-Request (RIR) from HSS

This procedure is used between the HSS and the SCEF, whenever HSS needs to send a report in RIR.

When the procedure is invoked by the HSS, it is used for reporting the:

- Change of association of the UE and UICC and/or new IMSI-IMEI-SV;
- UE reachability for SMS; and
- Roaming status (Roaming or No Roaming) of the UE, and change in roaming status of the UE.

It is also used to:

- Update the SCEF with the suspend/resume/cancel status of an ongoing monitoring. Only **Cancel** is supported for current SCEF release.
- Convey reports and/or status indications for all or some UEs belonging to a group.
- Indicate the reason of communication failure.
- Indicate the information when the UE transitions into idle mode.

For group based configuration processing, if the Group Guard Timer was included in the CIR command, the HSS sends the RIR command before the Group Guard Timer expires and includes several reports and/or status indications in one or more Group-Monitoring Event Report AVPs.

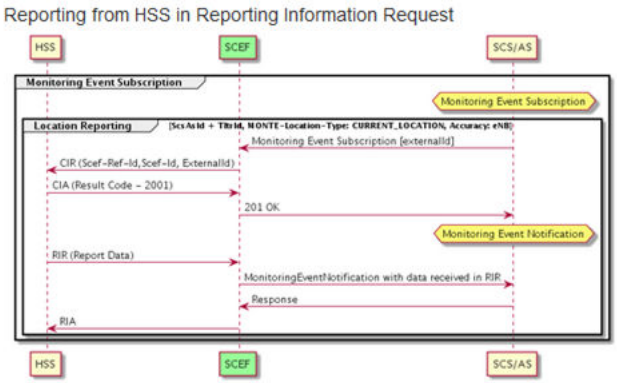
> **Note:**
>
> The HSS may divide the accumulated Monitoring Configuration Indications/ immediate reports into multiple messages.
> The HSS sends immediate reports and configuration indications for the group based configuration processing using the Group-Monitoring-Event-Report.

When the SCEF receives a RIR from the HSS, and at least one of the received Monitoring Event Reports has a SCEF-Reference-ID not known by the SCEF, it shall reply with DIAMETER_ERROR_SCEF_REFERENCE_ID_UNKNOWN. In that case, if the HSS had included multiple Monitoring Event Reports in the RIR command, the SCEF includes in the Reporting Information Answer command a list of Monitoring-Event-Report-Status AVPs where the status of multiple monitoring event reports is detailed. In that AVP list, the AVPs corresponding to event reports with a successful status may be omitted by the SCEF for efficiency.

SCEF compares the Monitoring type and its value received in message with the context. If there is any mismatch, it replies with DIAMETER_ERROR_SCEF_REFERENCE_ID_UNKNOWN. Otherwise, when the SCEF receives a RIR from the HSS, the SCEF sets the Experimental-Result to DIAMETER_SUCCESS in the Reporting Information Answer and handles it according to the procedures defined in 3GPP TS 23.682 Architecture enhancements to facilitate communications with packet data networks and applications.

For each successful report data in Group-Monitoring-Event-Report and the Monitoring Event Report AVPs, SCEF sends an HTTP post notification message to SCS/AS with details as received in RIR message.

Figure 5-2 Reporting from HSS



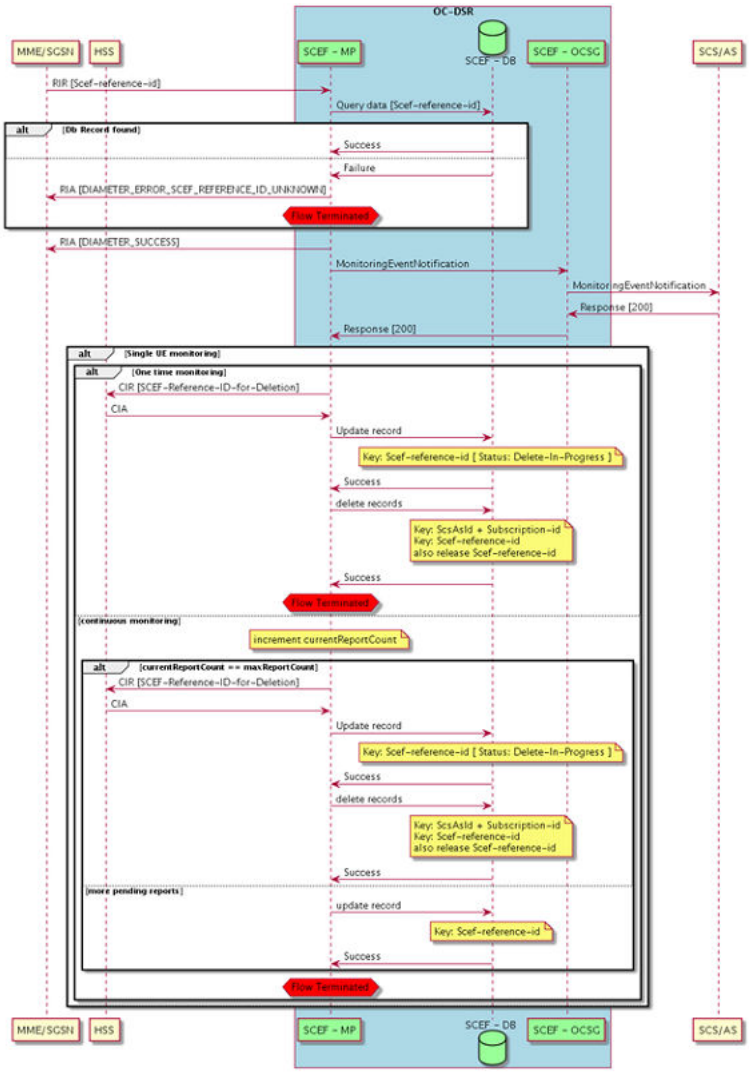
Notification in Reporting-Information-Request (RIR) from MME/SGSN

When the SCEF receives a Reporting Information Request from the MME/SGSN and at least one of the Monitoring Event Report AVPs has a SCEF-Reference-ID not known by the SCEF, it replies with Experimental-Result-Code set to DIAMETER_ERROR_SCEF_REFERENCE_ID_UNKNOWN. In that case, if the HSS had included multiple Monitoring Event Reports in the RIR command, the SCEF includes in the Reporting Information Answer command a list of Monitoring-Event-Report-Status AVPs where the status of multiple monitoring event reports is detailed. In that AVP list, the AVPs corresponding to event reports with a successful status may be omitted by the SCEF, for efficiency; otherwise, when the SCEF receives a Reporting-Information-Request command from the MME/SGSN, the SCEF sets Result-Code to DIAMETER_SUCCESS in the Reporting-Information-Answer and handles it according to the procedures defined in 3GPP TS 23.682 Architecture enhancements to facilitate communications with packet data networks and applications.

SCEF compares the Monitoring type, User Identifier, and its value received in message with the context. If there is any mismatch, it replies with DIAMETER_ERROR_SCEF_REFERENCE_ID_UNKNOWN.

For each successful report data in the Monitoring Event Report AVPs, SCEF sends an HTTP post notification message to SCS/AS with details as received in RIR message.

Figure 5-3 HTTP Post Notification



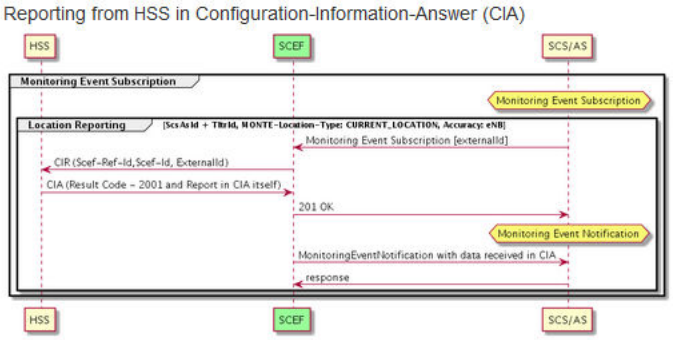
Notification in Configuration-Information-Answer (CIA)

This procedure is used between the HSS and the SCEF. HSS can send an available report for the Monitoring Event for the subscription done in the Monitoring Event Report AVPs in the Configuration-Information-Answer (CIA) message itself.

In case of a single report, for a successful report data in the Monitoring Event Report AVP, SCEF sends the report as a part of the Monitoring Subscription Response message.

In case of multiple reports, for each successful report data in the Monitoring Event Report AVPs, SCEF sends an HTTP post notification message to SCS/AS with details as received in the RIR message.

Figure 5-4 Reporting HSS



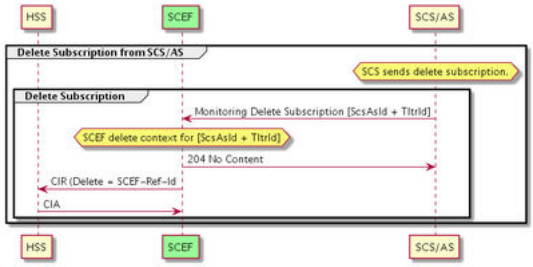
Monitoring Event Deletion Initiated from SCS/AS

SCS/AS can send HTTP message using Individual Monitoring Event Subscription and DELETE method. SCS/AS includes the subscription ID in URI, which needs to be deleted.

SCEF finds and deletes the context for Monitoring Event Subscription corresponding to SCS/AS and subscription ID received in HTTP message.

SCEF also sends Configuration-Information-Request (CIR) for deletion for SCEF Reference ID corresponding to SCS/AS and subscription ID received in the HTTP message.

Figure 5-5 Configuration-Information-Request

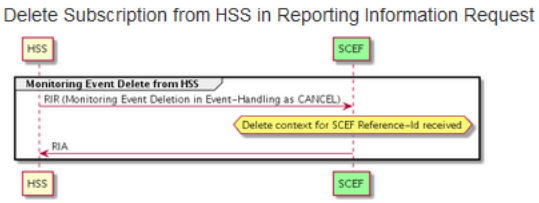


Monitoring Event Deletion Initiated from HSS

When a subscriber is deleted from the HSS while monitoring is active or the authorization for monitoring is revoked, the HSS sends an RIR command to the SCEF with the Event-Handling AVP set to the value CANCEL.

SCEF finds and deletes the context for Monitoring Event Subscription corresponding to SCEF Reference ID received in RIR message from HSS.

Figure 5-6 Delete Subscription from HSS



Monitoring Event Get

SCS/AS can send an HTTP message using the Individual Monitoring Event Subscription and GET method. SCS/AS includes the subscription ID in URI, which needs to be fetched.

SCEF finds and gets back the context data stored for the Monitoring Event Subscription corresponding to SCS/AS and subscription ID received in the HTTP message.

6
SCEF MMI Attributes

Access Control Associations

The Access Control Associations are used to associate the Rules with an Access Control List.

Table 6-1 Access Control Associations Attribute Details

Attribute	Type	Mandatory (Yes/No)
aclName	string	Yes
ruleName	string	Yes

Access Control Lists

The Access Control Lists are a set of configurations to support multi tenancy Access Control, DRMP, and Flow/Overload Control during notification callback.

Table 6-2 Access Control Lists Attribute Details

Attribute	Type	Mandatory (Yes/No)
name	string	Yes

Access Control Rules

Access Control Rules are a set of configurations to create an Access Control Rule.

Table 6-3 Access Control Rules Attribute Details

Attribute	Type	Enum Values	Mandatory (Yes/No)
name	string		Yes
userIdentifierType	string	MSISDN DOMAIN	Yes
startAddr	string		No
endAddr	string		No
domain	string		No
supportedFeatures[1]	string	Nidd Monitoring Triggering ECR	No

[1]Multiple values can be specified.

APN Configuration Sets

APN Configuration Sets are a set of configurations associated to an APN; the APN is associated to a SCS/AS Profile.

Table 6-4 APN Configuration Sets Attribute Details

Attribute	Type	Min/Max	Enum Values	Default Value	Mandatory (Yes/No)
name	string				Yes
value	string				Yes
maxPacketSize	integer	1/2500		100	No
maxPacketBufferSize	integer	1/1500		100	No
downlinkApnRateControlUnit	string		Unrestricted Minute Hour Day Week	Unrestricted	No
downlinkApnRateControlVal[1]	integer	0/60			No
downlinkApnMessageRate[1]	integer	1/1000			No
uplinkApnRateControlUnit	string		Unrestricted Minute Hour Day Week	Unrestricted	No
uplinkApnRateControlVal[1]	integer	0/60			No
uplinkApnMessageRate[1]	integer	1/1000			No
reliableDataService	Boolean		True	False	Yes (If testing the RDS feature)

[1]Has an unconfigured value of -1.

> **Note:**
>
> If the value of reliableDataService in the "ScefApnCfgSet" table changes from true to false or vice versa, the table will not have any effect on NiddContext. The NiddContext must be created with new RDS flag value using PATCH operation.

Device Triggering Configuration Sets

Device Triggering Configuration Sets are a set of configurations for Device Triggering to create a SCS/AS.

Table 6-5 Device Triggering Configuration Sets Attribute Details

Attribute	Type	Min/Max	Enum Values	Default Value	Mandatory (Yes/No)
name	string				Yes
defaultApplicationPort	integer	0/65535		1000	No
defaultPriority	string		NonPriority Priority	NonPriority	No
maxValidityPeriod	integer	-1/86400		3600	No
mandateApplicationPort	boolean		true false	false	No
mandatePriority	boolean		true false	false	No

Monitoring Event Configuration Sets

Monitoring Event Configuration Sets are a set of configurations for SCEF Monitoring Events.

Table 6-6 Monitoring Event Configuration Sets Attribute Details

Attribute	Type	Min/Max	Enum Values	Default Value	Mandatory (Yes/No)
name	string				Yes
monitoringType[2]	string		LossOfConnectivity UEReachability LocationReporting ChangeOfImsiImeiAssociation RoamingStatus CommunicationFailure AvailabilityAfterDdnFailure NumberOfUEsInAnArea		Yes
groupedMonitoringEnabled	boolean		true false	false	No
numberOfReports	integer	1/5000		1	No
maxMonitoringDuration	integer	1/86400		3600	No
maxUePerReport	integer	1/100		50	No
initiateRefDelete	boolean		true false	false	No
minTauRauTimerValue[1]	integer	0/2147483647			No
maxTauRauTimerValue[1]	integer	0/2147483647			No

Table 6-6 (Cont.) Monitoring Event Configuration Sets Attribute Details

Attribute	Type	Min/Max	Enum Values	Default Value	Mandatory (Yes/No)
minResponseTime[1]	integer	0/2147483647			No
maxResponseTime[1]	integer	0/2147483647			No
downlinkPackets	integer	1/100			No
enforceReportingInterval	boolean		true false	false	No
minReportingInterval[1]	integer	1/86400		120	No
locationAccuracy[3]	string		CgiEcgi eNB LaTaRa PLMN	eNB	No

[1]Has an unconfigured value of -1.

[2]User can select multiple values. Bit values are shown in Table 6-7.

[3]PLMN - Future Plan

Table 6-7 Bit Values

Enum	Bit Value
LossOfConnectivity	1
UEReachability	2
LocationReporting	4
ChangeOfImsiImeiAssociation	8
RoamingStatus	16
CommunicationFailure	32
AvailabilityAfterDdnFailure	64
NumberOfUEsInAnArea	128
EnhancedCoverage	256

Monitoring Location Areas

A set of configurations that apply to Monitoring Location Area call flows.

Table 6-8 Monitoring Location Areas

Attribute	Type	Min/Max	Enum Values	Default Value	Mandatory (Yes/No)
mcc	integer	0/999			Yes
mnc	integer	0/999			Yes
locationAreaCode	integer	0/65535			Yes
routingAreaCodeStart	integer	0/255			Yes

Table 6-8 (Cont.) Monitoring Location Areas

Attribute	Type	Min/Max	Enum Values	Default Value	Mandatory (Yes/No)
routingAreaCodeEnd	integer	0/255			Yes
trackingAreaCodeStart	integer	0/42949672			Yes
trackingAreaCodeEnd	integer	0/42949672			Yes
poolName	string				Yes
locationAreaFqdn	string				Yes
locationAreaRealm	string				Yes
locationAreaType	string		TRACKINGAREAID ROUTINGAREAID		Yes

NIDD Configuration Sets

NIDD Configuration Sets are a set of configurations that apply to NIDD call flows.

Table 6-9 NIDD Configuration Sets Attribute Details

Attribute	Type	Min/Max	Enum Values	Default Value	Mandatory (Yes/No)
name	string				Yes
dataDuration	integer	0/2678400		0	No
maxAuthDuration	integer	60/ 1000000000		86400	No
maxOnholdDataMsg	integer	1/5		1	No
minRetransmissionTime	integer	0/10		5	No
pdnEstablishmentOptionEnabled	boolean		true false	false	No
pdnEstablishmentOption	string		WAIT_FOR_UE INDICATE_ERROR	INDICATE_ERROR	No
priority	integer	0/15		0	No

SCS/AS

SCS/AS enables applications to access and use functionality provided by Service Components over standardized interfaces (APIs).

Table 6-10 SCS/AS Attribute Details

Attribute	Type	Min/Max	Enum Values	Default Value	Mandatory (Yes/No)
scsAsId	string				Yes
niddCfgSetName	string				No
apnCfgSetName	string				No

Table 6-10 (Cont.) SCS/AS Attribute Details

Attribute	Type	Min/Max	Enum Values	Default Value	Mandatory (Yes/No)
monitoringEventCfgSetName	string				No
deviceTriggeringCfgSetName	string				No
callbackUrl	string				No
isEcrAllowed	boolean		true false	false	No
interimInterval	integer	10/3600		600	No
smsScFqdn[1]	string				No
smsScRealm[1]	string				No
scsAsIsdn[1]	string				No
aclName[2]	string				No

[1]Applicable only when Device Triggering is configured .

[2]Related to Access Control

System Options

SCEF related user configurable Options.

Table 6-11 System Options Attribute Details

Attribute	Type	Min/Max	Enum Values	Default Value	Mandatory (Yes/No)
art	string				No
prt	string				No
apiGwIpList	string				Yes
priority	integer	0/15		0	No
retryDbUpdate	integer	0/5		2	No
servingPlmnRateControlEnabled	boolean		true false	false	No
longestSubdomainMatchEnabled	boolean		true false	false	No
scefWaitTime	integer	1/180		1	No
scefId[1]	string				No
binaryEncoder	string		ASCII Base2 Base16 Base64	Base64	No

[1]This maps to the FQDN of Local Node

7

Device Status Query Troubleshooting API

This section describes how to use Service Capability Exposure Function (SCEF) Device Status Query Troubleshooting Application Program Interface (API) and access REST interface URLs.

Device Status Query Troubleshooting API Overview

Device Status Query Troubleshooting Application Program Interface (API) can retrieve and delete IP and non-IP data corresponding to a device from Service Capability Exposure Function (SCEF).

Device Status Query Troubleshooting API provides the following APIs:

- Non-IP Data Delivery (NIDD) Troubleshooting API
- Monitoring Event (MONTE) Troubleshooting non-IP API
- Monitoring Event (MONTE) Troubleshooting IP API

Understanding the APIs

You can access these APIs using any existing operator's account that is used to log in to the Oracle Communications Service Gatekeeper (OCSG) Partner Relationship Management (PRM) portal. A new operator account can be created through the Weblogic Admin portal.

For creating users, refer to Create User Account.

NIDD Troubleshooting API

This API accepts the following query parameters to retrieve the data:

- externalId
- msisdn
- imsi
- tltrId

At least one query parameter is required to process the request. If more than one query parameter is present in the request, then the query parameters are processed in the following order of precedence:

1. tltrId
2. externalId
3. msisdn
4. imsi

For example, if all the four query parameters are present in the request, the API considers the **tltrId** parameter first to process the request. If the **tltrId** parameter is not provided, then API considers the next available parameter according to the order of precedence.

This API supports GET and DELETE methods.

Retrieving and Deleting Data

URL Format

https://<APPServer IP>:9002/troubleshooting/oam/nidd?<parameter=input value>

where, <parameter=input value> indicates the parameter name and its value.

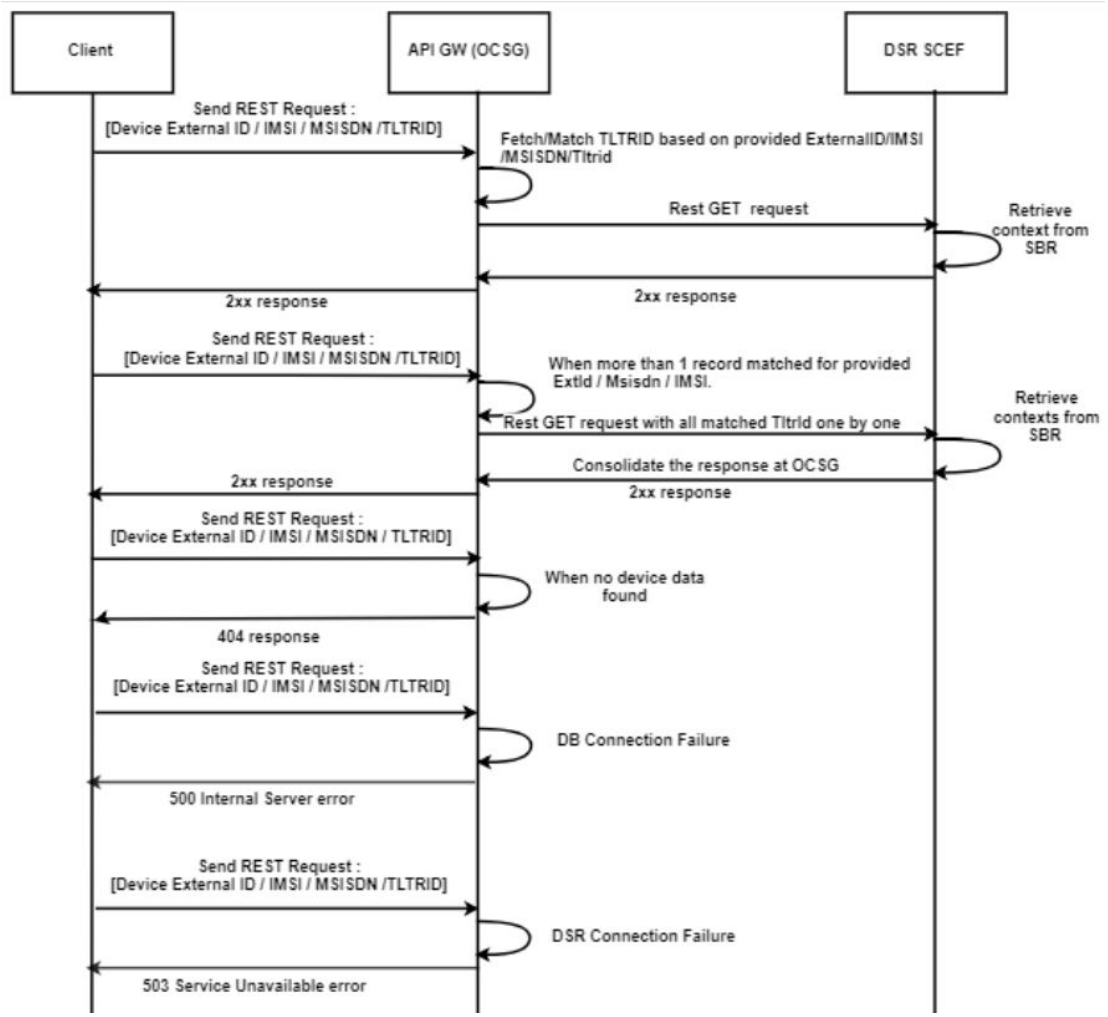
Example of REST interface URLs

- https://<APPServer IP>:9002/troubleshooting/oam/nidd?tltrId=5e66ed330f3200470000000d
- https://<APPServer IP>:9002/troubleshooting/oam/nidd?externalId=extid1@oracle.com
- https://<APPServer IP>:9002/troubleshooting/oam/nidd?msisdn=7799383911
- https://<APPServer IP>:9002/troubleshooting/oam/nidd?imsi=123456789123456

GET Call Flow

This call flow explains different scenarios involved in retrieving non-IP delivery data corresponding to a device.

Figure 7-1 NIDD GET Call Flow



Sample Request

https://10.75.225.37:9002/troubleshooting/oam/nidd?msisdn=555555555

Sample Respose


```
{
  "niddDeviceInfo": [
    {
      "tltrId": "5f081abb01ca00ce00000006",
      "status": "Success",
      "data": {
        "version": 1,
        "recordStatus": "Created",
        "scsAsId": "SCSAS50",
        "imsi": "1234512351",
        "grantTime": "03/15/52 13:29:43",
        "apn": "apn50.test.com",
        "epsBearerId": "Not Set",
```

```
        "msisdn": "555555555",
        "mmeSgsnFqdn": "Not Set",
        "plmnId:mcc": 0,
        "plmnId:mnc": 0,
        "imei": "Not Set",
        "imeiSv": "Not Set",
        "ratType": 4294967295,
        "retransmissionTime": "Not Set",
        "ueAvailability": false,
        "servingPlmnRate": 0,
        "currentDownlinkApnRate": 0,
        "currentUplinkApnRate": 0,
        "currentPlmnRate": 0,
        "downlinkApnRateResetTime": "12/31/69 19:00:00",
        "uplinkApnRateResetTime": "12/31/69 19:00:00",
        "plmnRateResetTime": "12/31/69 19:00:00",
        "notificationUri": "http://10.75.224.65:10001/scs/resources/
t8",
        "pdnEstablishmentOption": "WAIT_FOR_UE",
        "pdnConnectionTime": "12/31/69 19:00:00",
        "List of Pending Data": [
                    {
                              "ttrId": "1",
                              "data": "Hello from Network10",
                              "size": 20,
                             "priority": 1,
                             "maximumLatency": "08/12/20 14:00:00"
                    }
         ]
      }
    }
  ]
}
```

Note:

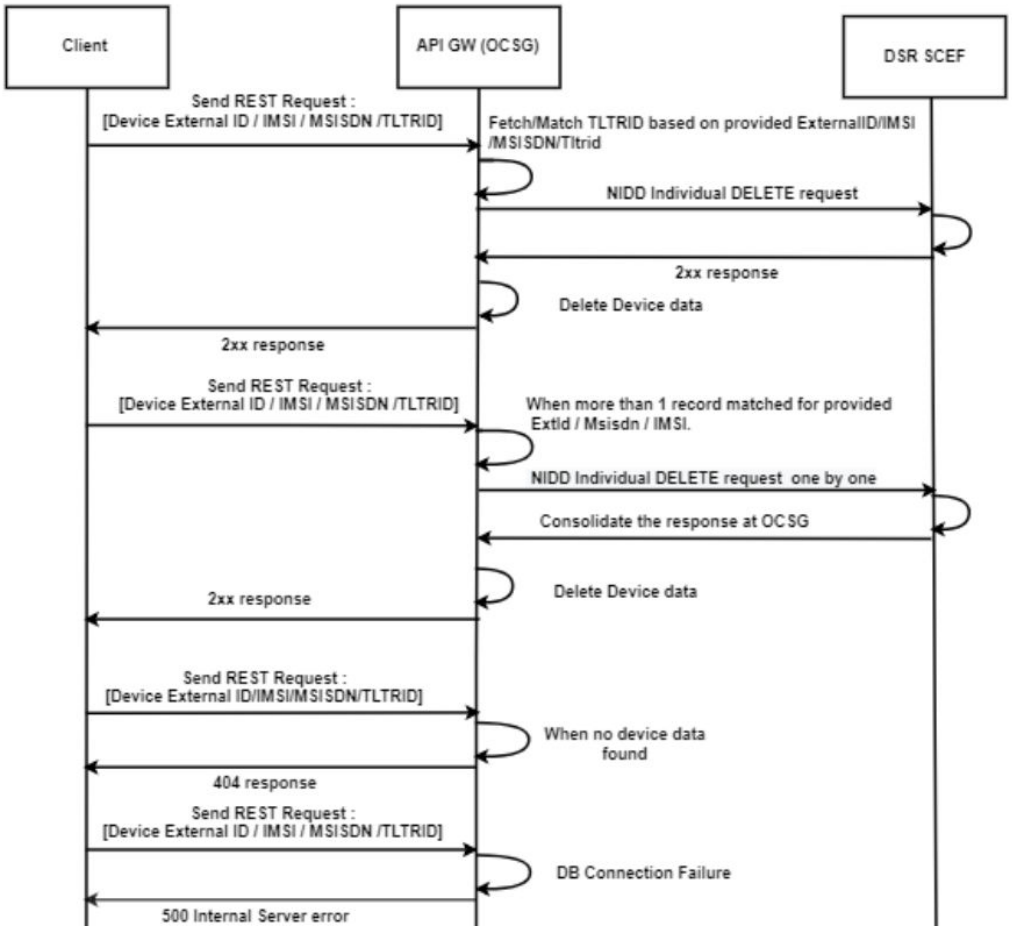
The following are some default values for the NIDD device information:

```
"mmeSgsnFqdn": "Not Set",
"plmnId:mcc": 0,
"plmnId:mnc": 0,
"imei": "Not Set",
"imeiSv": "Not Set",
"ratType": 4294967295,
"retransmissionTime": "Not Set",
"ueAvailability": false,
"servingPlmnRate": 0,
"currentDownlinkApnRate": 0,
"currentUplinkApnRate": 0,
"currentPlmnRate": 0,
"downlinkApnRateResetTime": "12/31/69 19:00:00",
"uplinkApnRateResetTime": "12/31/69 19:00:00",
"plmnRateResetTime": "12/31/69 19:00:00",
"recordStatus" could have the following values - Pending, Created or
DeleteInProgress
```

DELETE Call Flow

This call flow explains different scenarios involved in deleting the non-IP delivery data corresponding to a device.

Figure 7-2 NIDD DELETE Call Flow



Sample Request

https://10.75.225.37:9002/troubleshooting/oam/nidd?msisdn=555555555

Sample Response

```
{
    "niddDeleteResult": [
        {
            "status": "200 OK",
            "tltriId": "5f081abb01ca00ce00000006"
        }
    ]
}
```

> **Note:**
>
> When several contexts exist for a given device, response is provided with all Tltrid's and their operation status for each.

MONTE Troubleshooting Non-IP API

This API accepts the following query parameters to retrieve the data:

- externalId
- msisdn
- tltrId

At least one query parameter is required to process the request. If more than one query parameter is present in the request, then the query parameters are processed in the following order of precedence:

1. tltrId
2. externalId
3. msisdn

For example, if all the three query parameters are present in the request, the API considers the **tltrId** parameter first to process the request. If the **tltrId** parameter is not provided, then API considers the next available parameter according to the order of precedence.

This API supports GET and DELETE methods.

Retrieving and Deleting Data

URL Format

https://<APPServer IP>:9002/troubleshooting/oam/me-nonip?<parameter=input value>

where, <parameter=input value> indicates the parameter name and its value.

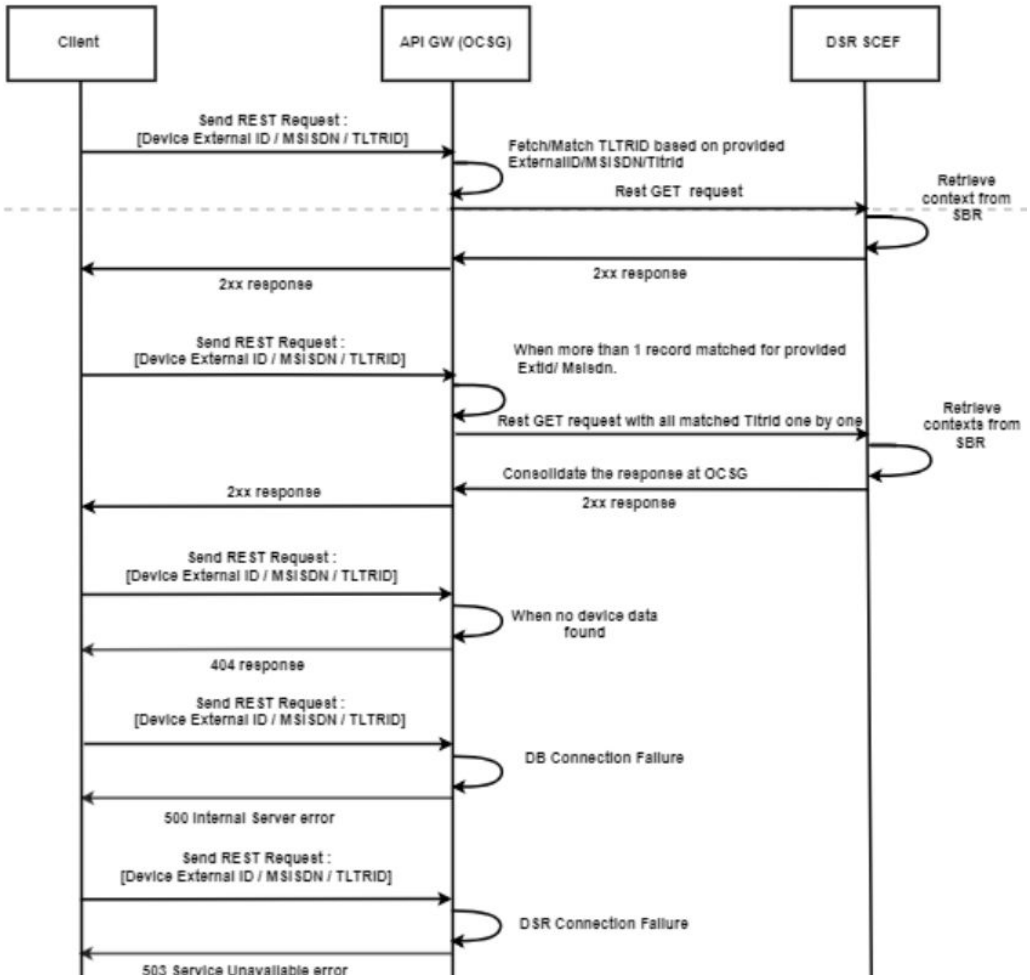
Example of REST Interface URLs:

- https://<APPServer IP>:9002/troubleshooting/oam/me-nonip?tltrId=5e66ed330f3200470000000d
- https://<APPServer IP>:9002/troubleshooting/oam/me-nonip?externalId=extid1@oracle.com
- https://<APPServer IP>:9002/troubleshooting/oam/me-nonip?msisdn=7799383911 3.2.2

GET Call Flow

This call flow explains different scenarios involved in retrieving non-IP delivery data corresponding to a device.

Figure 7-3 MONTE Non-IP GET Call Flow



Sample Request

https://10.75.225.37:9002/troubleshooting/oam/me-nonip?msisdn=1234567890

Sample Response

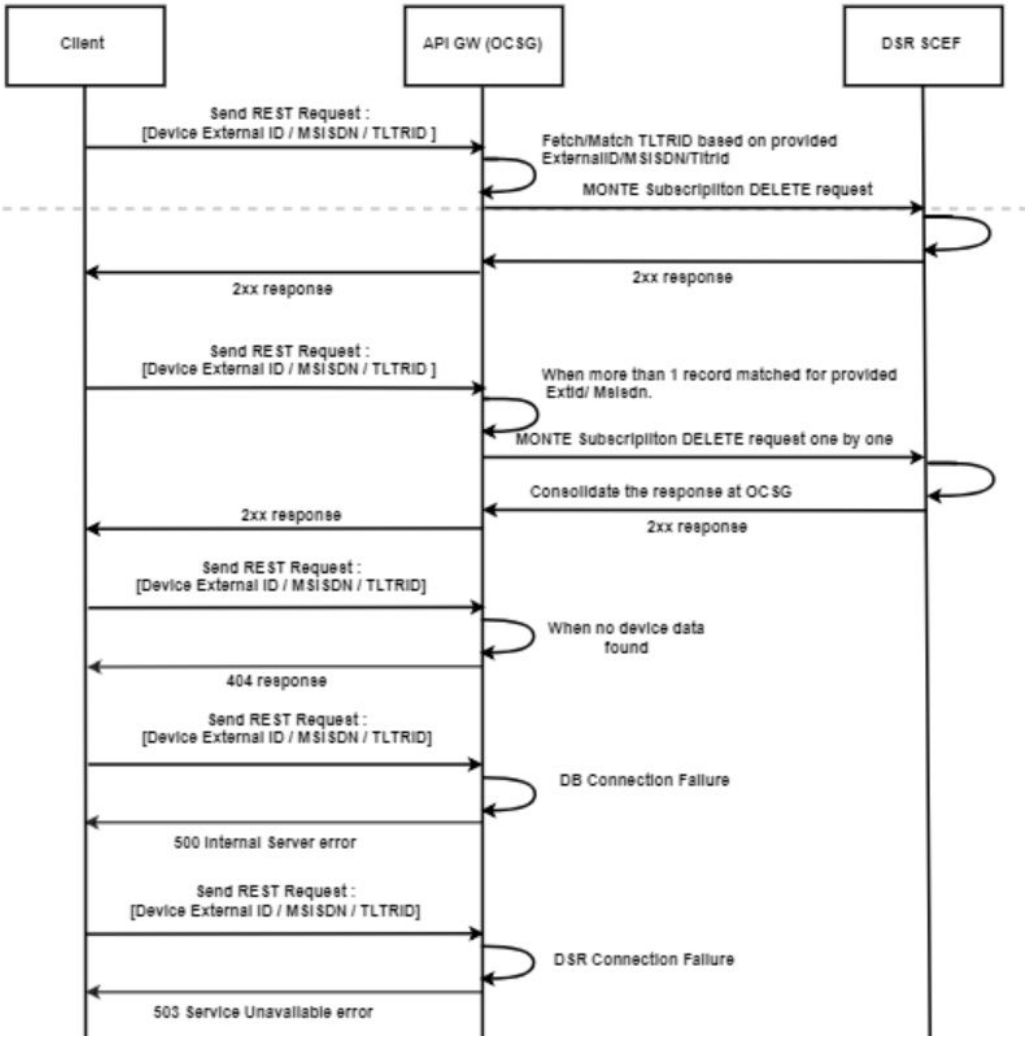
```
{
  "monitoringDeviceInfo": [
    {
      "subscriptionId": "5f0822a801ca00ce0000000a",
      "status": "Success",
      "data": {
        "version": 1,
        "recordStatus": "Created",
        "scefReferenceId": 10,
        "scsAsId": "SCSAS50",
        "notificationUri": "http://10.10.10.6:8787/configurations/
notification",
        "monitoringType": "UE_REACHABILITY",
        "reachabilityType": "SMS",
```

```
        "maxLatencyPresent": false,
        "maxRespTimePresent": false,
        "numDlPacketsPresent": false,
        "monitorExpireTime": "Not Set",
        "maxAllowedReportCount": 1,
        "currentReportCount": 0,
        "userIdentityInUse": "ExternalIdentifier",
        "externalId": "56789112@StreetLight-BLR-22.com",
        "msisdn": "988739322222",
        "hssRealm": "oracle.com",
        "hssFqdn": "peer1.oracle.com"
      }
    },
    {
      "subscriptionId": "5f08241501ca00ce0000000d",
      "status": "Success",
      "data": {
        "version": 1,
        "recordStatus": "Created",
        "scefReferenceId": 13,
        "scsAsId": "SCSAS70",
        "notificationUri": "http://10.75.225.37:10001/configurations/
notification",
        "monitoringType": "AVAILABILITY_AFTER_DDN_FAILURE",
        "idleStatusIndicationPresent": true,
        "monitorExpireTime": "Not Set",
        "maxAllowedReportCount": 100,
        "currentReportCount": 10,
        "userIdentityInUse": "ExternalIdentifier",
        "externalId": "111111111111@StreetLight-BLR-10.com",
        "hssRealm": "oracle.com",
        "hssFqdn": "peer1.oracle.com"
      }
    }
  ]
}
```


DELETE Call Flow

This call flow explains different scenarios involved in deleting the non-IP delivery data corresponding to a device.

Figure 7-4 MONTE Non-IP DELETE Call Flow



Sample Request

https://10.75.225.37:9002/troubleshooting/oam/me-nonip?msisdn=1234567890

Sample Response

```
{
    "monteDeleteResult": [
        {
            "status": "204 No Content",
            "subscriptionId": "5f0822a801ca00ce0000000a"
        },
        {
            "status": "204 No Content",
            "subscriptionId": "5f08241501ca00ce0000000d"
        }
    ]
}
```

> **Note:**
>
> When several contexts exist for a given device, response is provided with all Tltrid's and their operation status for each.

MONTE Troubleshooting IP API

This API accepts the following query parameters to retrieve the data:

- externalId
- msisdn
- imsi
- ip
- subscriptionid

At least one query parameter is required to process the request. If more than one query parameter is present in the request, then the query parameters are processed in the following order of precedence:

1. externalId
2. msisdn
3. imsi
4. ip
5. subscriptionid

For example, if all the five query parameters are present in the request, the API considers the **externalId** parameter first to process the request. If the **externalId** parameter is not provided, then API considers the next available parameter according to the order of precedence.

> **Note:**
>
> The **subscriptionid** parameter mentioned in the query parameter must be of only application subscriptions. Device **SubscriptionId** is not accepted.

This API supports GET and DELETE methods.

Retrieving and Deleting Data

In MONTE IP GET, when subscriptionid is provided as an input query parameter, then the device of that subscription is searched, and subscriptions data, such as App subscriptions, Device subscriptions, and Buffered messages, are retrieved for the given device.

In MONTE IP DELETE, when subscriptionid is provided as an input query parameter, then the device of that subscription is searched, and subscriptions data, such as App subscriptions and Device subscriptions, are deleted for the given device.

URL Format

https://<APPServer IP>:9002/troubleshooting/oam/me-ip?<parameter=input value>

where <parameter=input value> indicates the parameter name and its value.

Examples of REST interface URLs

- https://<APPServer IP>:9002/troubleshooting/oam/me-ip?externalId=extid1@oracle.com
- https://<APPServer IP>:9002/troubleshooting/oam/me-ip?msisdn=7799383911
- https://<APPServer IP>:9002/troubleshooting/oam/me-ip?imsi=7799383911
- https://<APPServer IP>:9002/troubleshooting/oam/me-ip?ip=10.75.224.61
- https://<APPServer IP>:9002/troubleshooting/oam/me-ip?subscriptionid=046b6c7a-0b8b-43b9-b35c-6489e6daee92

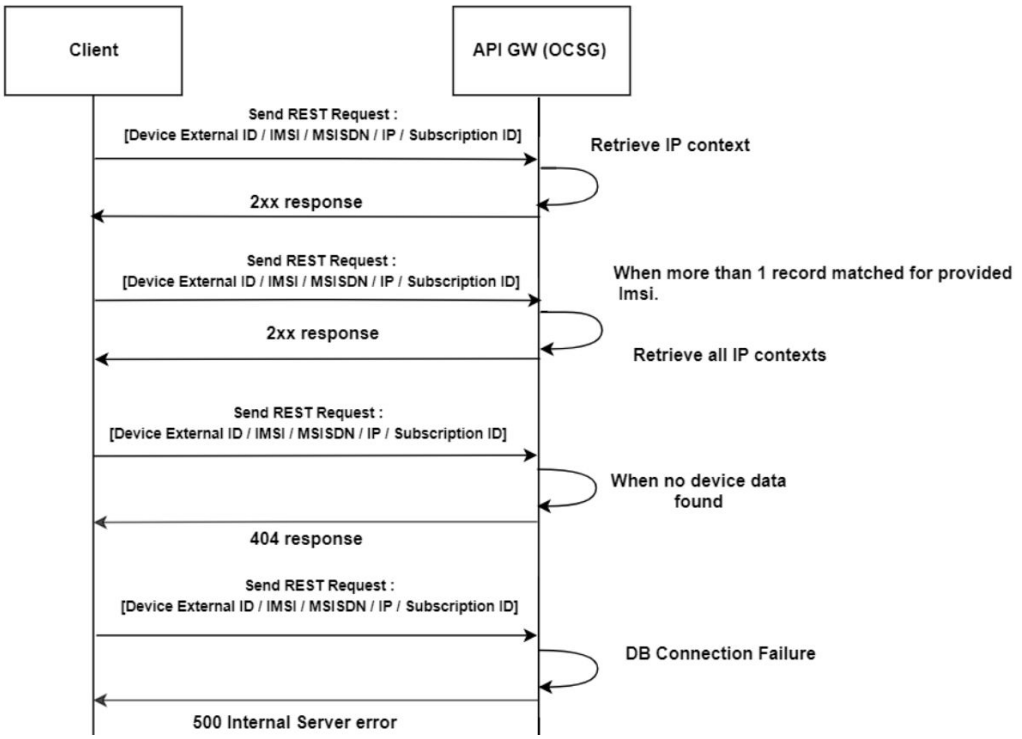
GET Call Flow

Table 7-1 Device Status

Status	Description
Not Connected	Device is never connected to MQTT Broker
Connected	Device is connected to MQTT Broker
Disconnected	Device is not connected to MQTT Broker

This call flow explains different scenarios involved in retrieving IP data corresponding to a device.

Figure 7-5 MONTE IP GET Call Flow



Sample Request

https://10.75.225.37:9002/troubleshooting/oam/me-ip?msisdn=7829139111

Sample Response

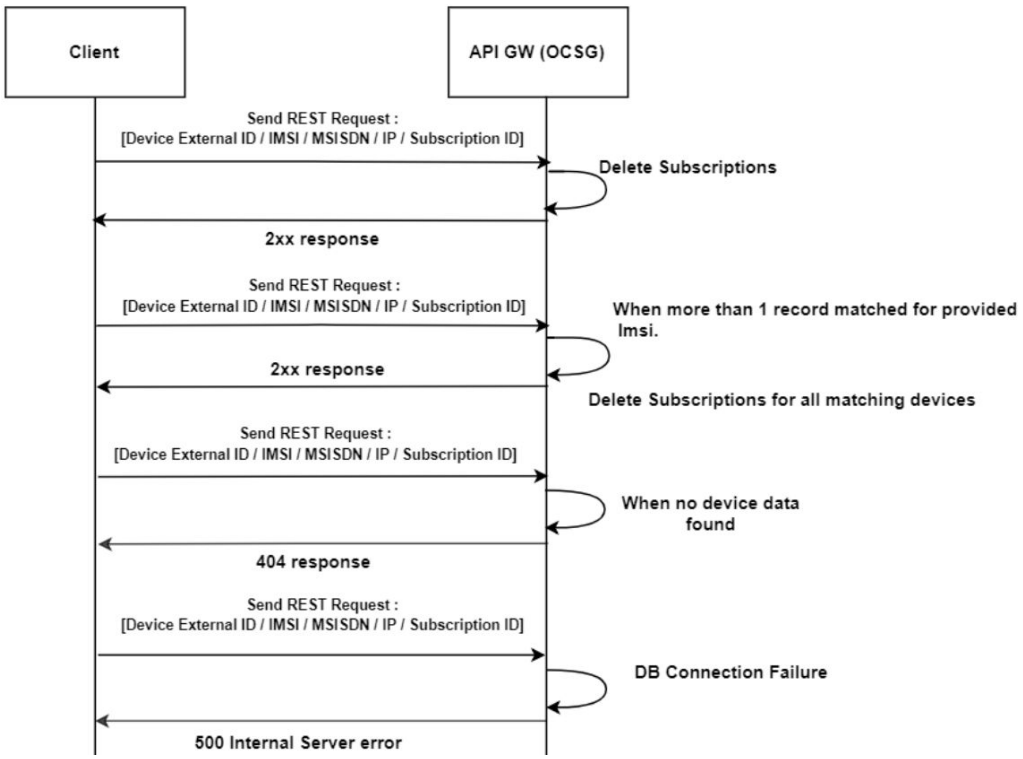

```
{
    "ipDevices": [
        {
            "apn": "test.ashok3.com",
            "applicationSubscription": [
                {
                    "appId": "partner1-scefapp",
                    "deviceInfo": "7829139111",
                    "subscribedTopic": "tracking/batterylevel/#",
                    "subscriptionId": "b23a82eb-ef45-41fe-9004-e7c695731d92"
                }
            ],
            "bufferedMessages": [
                {
                    "bufferedData": "ashok sent at 10:58ampm on 07102020",
                    "publishedTopic": "tracking/batterylevel/ashok",
                    "subscribedTopic": "tracking/batterylevel/ashok"
                }
            ],
            "deviceId": "Ashok3",
            "deviceSubscription": [
                {
                    "subscribedTopic": "tracking/batterylevel/ashok",
                    "subscriptionId": "72739d0d-b89b-4b4c-8a29-b66c0ef9b7e6"
                }
            ],
            "deviceType": "MQTT",
            "externalId": "7829139111@StreetLight-BLR-5",
            "imsi": 7829139111,
            "msisdn": 7829139111,
            "sessionId": ".;1096298392;1",
            "status": "Disconnected"
        }
    ]
}
```


DELETE Call Flow

This call flow explains different scenarios involved in deleting the IP data corresponding to a device.

MONTE IP DELETE deletes the device subscriptions and application subscriptions. If the device is connected to the Message Queuing Telemetry Transport (MQTT) broker, it remains in the connected state.

Figure 7-6 MONTE IP DELETE Call Flow



Sample Request

https://10.75.225.37:9002/troubleshooting/oam/me-ip?msisdn=7829139111

Sample Response

200 OK

Device Status Query Troubleshooting API Configuration

Device Troubleshooting Query APIs expose Management bean that can be accessed through the OCSG Admin portal.

To modify the default values:

1. Navigate to the Management bean: **Service-gatekeeper-domain**, and then **OCSG**, and then **AppServerX**, and then **Communication Services**, and then **SCEFDTS_Configuration**.
   The following screen appears:

Figure 7-7 Management Bean

2. Change the default values as required.
3. Click on **Update Attributes**.

Table 7-2 Attribute Description

Name	Description	Scope
ReadTimeOut	Read time out in ms. The default value is 2000 ms.	Cluster, Shared across AppServers. Changes made on one AppServer reflects on all other AppServers.
ConnectTimeOut	Connect time out in ms. The default value is 30000 ms.	Cluster, Shared across AppServers. Changes made on one AppServer reflects on all other AppServers.

A

OCSG Introduction

This appendix describes how to configure Oracle Communications Services Gatekeeper (OCSG) and then provision it.

Custom Configuration

This section describes how to configure the OCSG.

Custom configuration of the OCSG involves these steps:

- Configure DSR MP IPs in DSR API GW
- Add SNMP Trap Receiver
- Change SNMP Version
- Generate MIB File
- Change General Logging Level
- Enable T8 Logging
- Change Statistics Storage Interval
- Enable CDRs
- Start/Restart Administrative Server
- Start/Restart Application Server
- Stop the Administrative and Application Servers
- Alarms
- Add New XSI to OCSG
- Change the Administrative Console Account Password
- Create User Account
- Change the Operator Account Password
- Purge Database Tables
- Set Up the Two-Way SSL Configuration, which includes:
  - Import Client Certificate
  - Import Server Certificate
- Change SSL Certificates and Private Keys

Configure DSR MP IPs in DSR API GW

Any change made to an Application Server (AppServer) reflects to all other servers.

1. Access the DSR API GW Admin console using https://<Admin-Server-XMI-IP>:9002/console.

2. Login using the admin account created when configuring the API GW.

   The default username is weblogic.

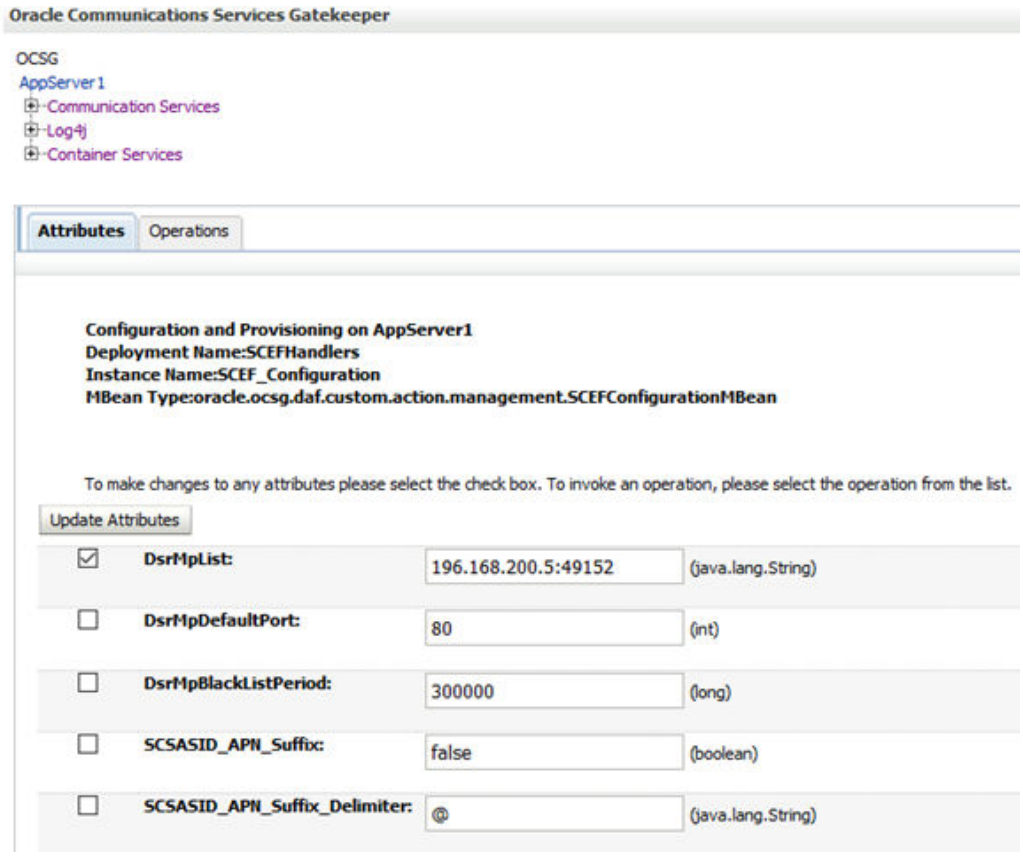
3. Navigate to **OCSG**, and then **AppServerx**, and then **Communication Services**, and then **SCEF_Configuration**, and then **Attributes (tab)**.

4. If the ports are different for each MP server, change the *DsrMpList* to a ip1:port,ip2:port,ip3port... format.

   If the IPs are in a ip1,ip2, ip3... format, then provide the port in the *DsrMpDefaultPort*.

5. Mark the associated checkbox and click **Update Attributes**.

Figure A-1 Configure Communication Services Attributes



Add SNMP Trap Receiver

This procedure is performed on each AppServer.

1. Access the DSR API GW Admin console using https://<Admin-Server-XMI-IP>:9002/console.

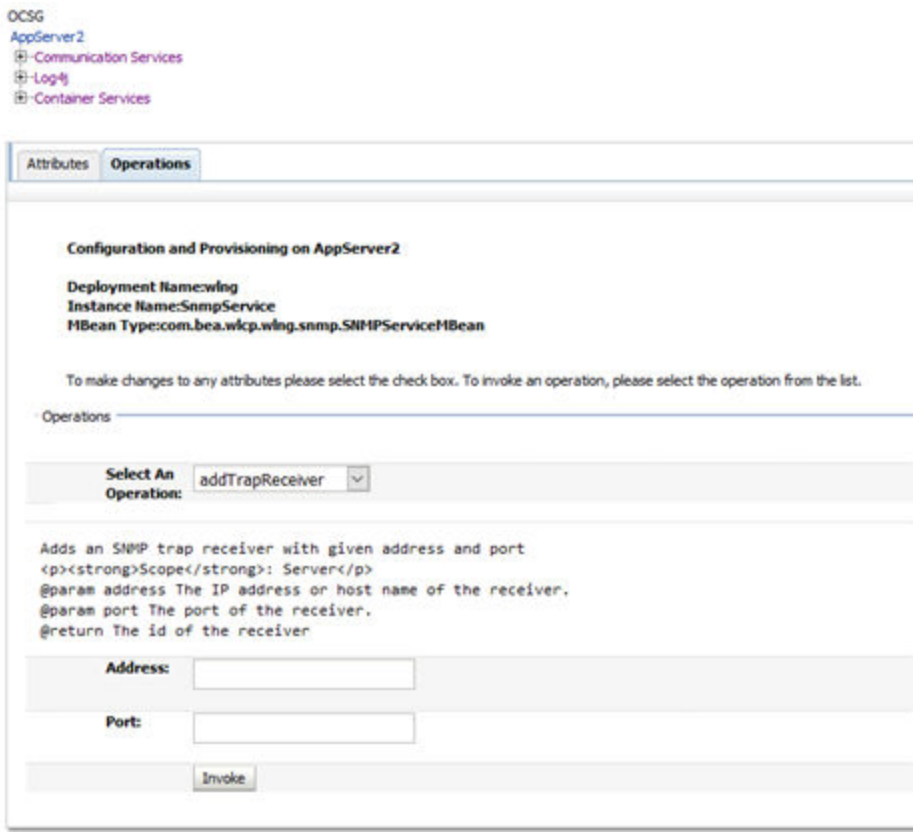
2. Login using the admin account created when configuring the API GW.

   The default username is weblogic.

3. Navigate to **OCSG**, and then **AppServerx**, and then **Container Services**, and then **SnmpService**, and then **Operations (tab)**.

4. For the *AddTrapReceiver* operation, enter the **Address** (IP address of the SNMP trap receiver) and **Port** (Port to which the SNMP traps should be sent) information.

Figure A-2 Add SNMP Trap



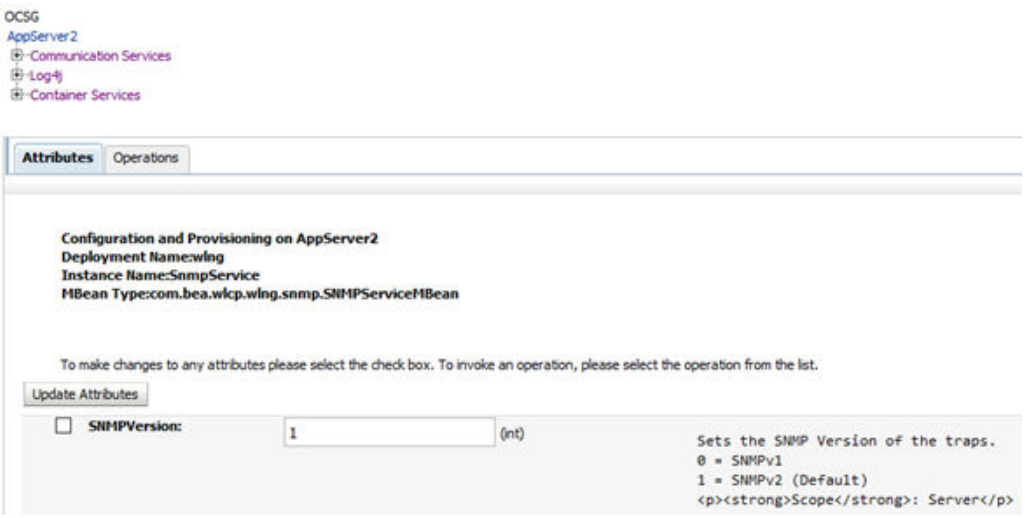
Change SNMP Version

This procedure is performed on each AppServer.

1. Access the DSR API GW Admin console using https://<Admin-Server-XMI-IP>:9002/console.
2. Login using the admin account created when configuring the API GW.

   The default username is weblogic.
3. Navigate to **OCSG**, and then **AppServerx**, and then **Container Services**, and then **SnmpService**, and then **Attributes (tab)**.
4. Change the *SNMPVersion* to 1 or 2.
5. Mark the associated checkbox and click **Update Attributes**.

Figure A-3 Change SNMP Version

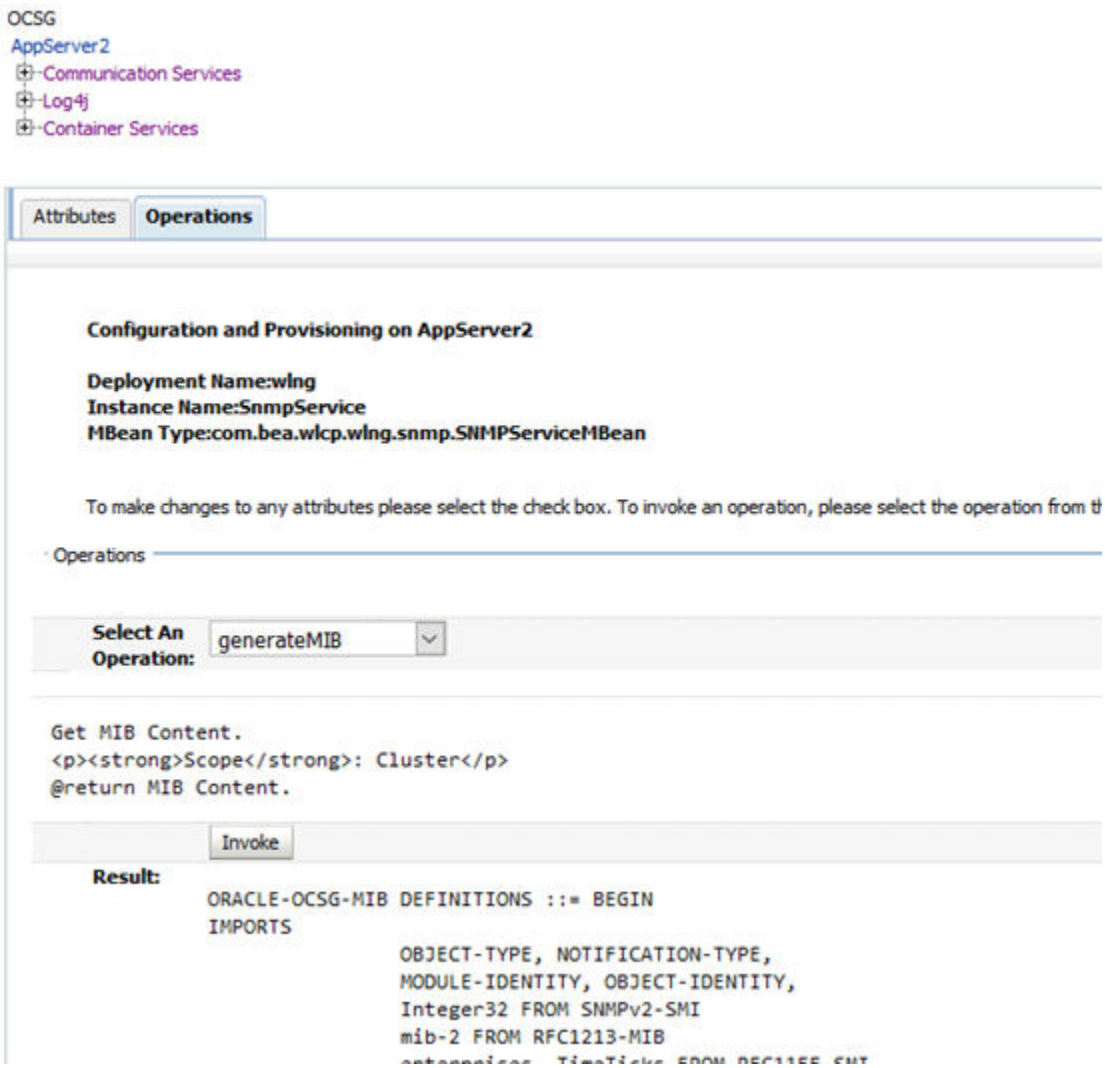


Generate MIB File

1. Access the DSR API GW Admin console using https://<Admin-Server-XMI-IP>:9002/console.
2. Login using the admin account created when configuring the API GW.

   The default username is weblogic.
3. Navigate to **OCSG**, and then **AppServerx**, and then **Container Services**, and then **SnmpService**, and then **Operations (tab)**.
4. Select the *generateMIB* operation and click **Invoke**.

Figure A-4 Generate MIB File



Change General Logging Level

This procedure is performed on each AppServer.

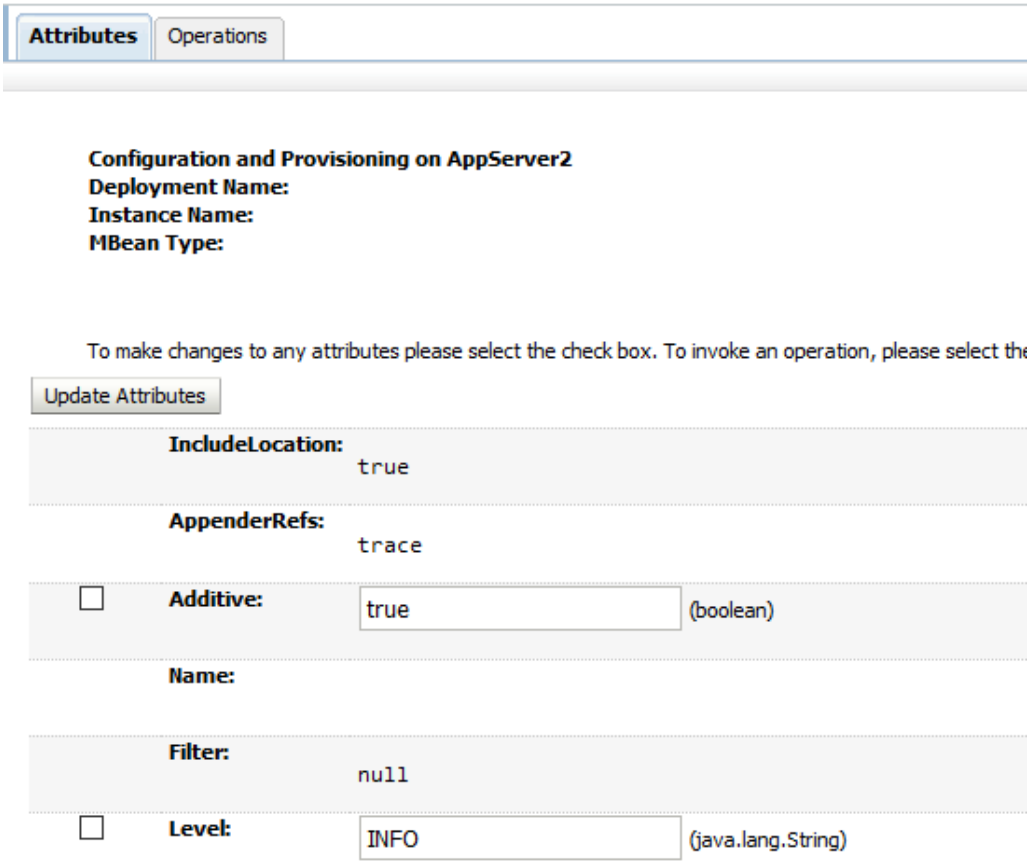
1. Access the DSR API GW Admin console using https://<Admin-Server-XMI-IP>:9002/console.
2. Login using the admin account created when configuring the API GW.

   The default username is weblogic.
3. Navigate to **OCSG**, and then **AppServerx**, and then **Log4J**, and then **log4j:Location=AppServerx,hierarchy=default,logger=root**, and then **Operations (tab)**.
4. Change the Level to one of these values
   - ALL
   - DEBUG

- ERROR
- FATAL
- INFO
- OFF
- TRACE
- WARN

5. Mark the associated checkbox and click **Update Attributes**.

Figure A-5 Change Log Level



Enable T8 Logging

T8 Logging can be enabled per API (NIDD/ME/DT/ECR) through DSR API GW Partner and API management Portal.

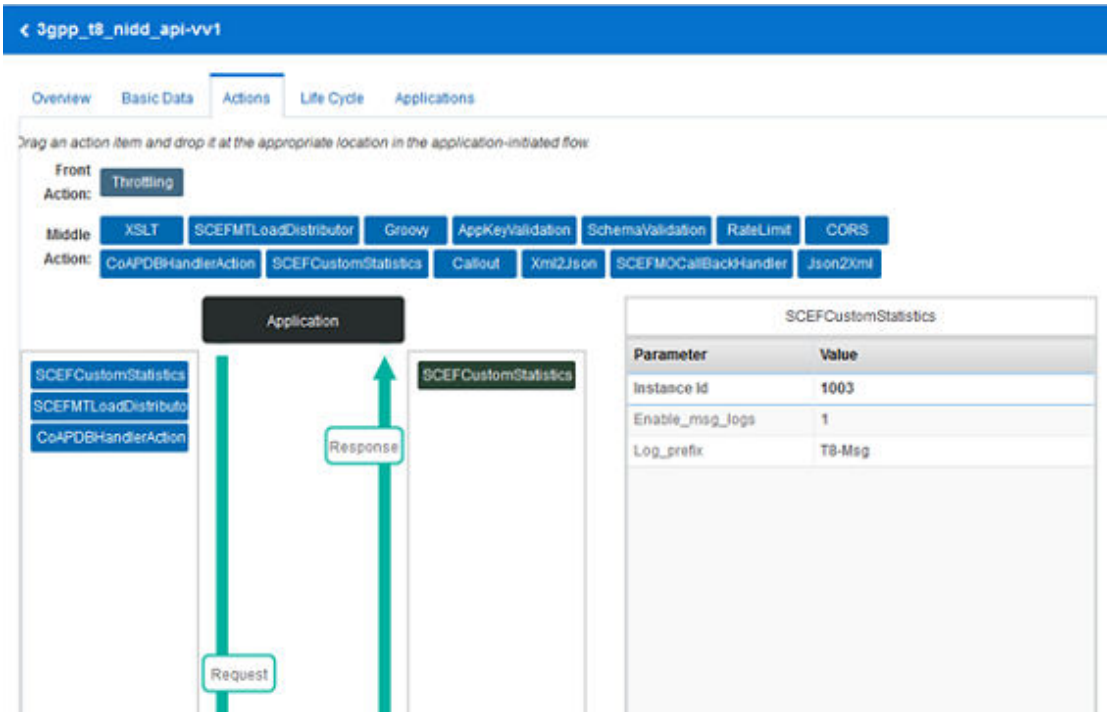
The T8 Request and Responses are logged into the /u03/app/oracle/ocsg-x.x.x/user_projects/domains/ services-gatekeeper-domain/scef/t8.log file.

1. Access PRM portal using https://<Appserverx-XMI-IP>:9002/portal/partner-manager/index/login.html url.
2. Login using the admin account created when configuring the API GW.

The default username is weblogic.

3. Click on APIs, and click on API for which T8 Logging should be enabled.
4. Select the Actions tab.
5. Select *SCEFCustomStatistics* on request path and update *Enable_msg_logs* to 1.
6. Click **Save**.
7. Repeat this procedure for the response path.
8. To disable T8 logging, repeat this procedure but change the *Enable_msg_logs* value to 0.

Figure A-6 Enable T8 Logging



Change Statistics Storage Interval

DSR API GW generates various types of SCEF-related statistics that exist within the DSR API GW database table as slee_statistics_data. Each statistics is represented by a numerical ID.

Table A-1 NIDD Statistics

Statistic Name	Statistic ID
NIDD Configuration create	
- Request received	1021
- Successful response sent	1022
- Error response sent	1023
NIDD Configuration read	

Table A-1 (Cont.) NIDD Statistics

Statistic Name	Statistic ID
- Request received	1026
- Successful response sent	1027
- Error response sent	1028
Individual NIDD Configuration request	
- Request received	1031
- Successful response sent	1032
- Error response sent	1033
Individual NIDD Configuration delete request	
- Request received	1036
- Successful response sent	1037
- Error response sent	1038
Individual NIDD configuration read request	
- Request received	1041
- Successful response sent	1042
- Error response sent	1043
NIDD Download Link data deliveries create	
- Request received	1046
- Successful response sent	1047
- Error response sent	1048
NIDD Download Link data deliveries read	
- Request received	1051
- Successful response sent	1052
- Error response sent	1053
Individual NIDD Download Link data deliveries create	
- Request received	1056
- Successful response sent	1057
- Error response sent	1058
Individual NIDD Download Link data deliveries read	
- Request received	1061
- Successful response sent	1062
- Error response sent	1063
NIDD Configuration update notification	
- Request sent	1066
- Successful response received	1067
- Error response received	1068
NIDD downlink data delivery status notification	
- Request sent	1071
- Successful response received	1072
- Error response received	1073
NIDD uplink data delivery status notification	
- Request sent	1076
- Successful response received	1077

Table A-1 (Cont.) NIDD Statistics

Statistic Name	Statistic ID
- Error response received	1078

Table A-2 Event Monitoring Statistics

Statistic Name	Statistic ID
Monitoring event subscriptions read	
- Request received	1111
- Successful response sent	1112
- Error response sent	1113
Monitoring event subscriptions create	
- Request received	1116
- Successful response sent	1117
- Error response sent	1118
Individual monitoring event subscriptions read	
- Request received	1121
- Successful response sent	1122
- Error response sent	1123
Individual monitoring event subscriptions create	
- Request received	1126
- Successful response sent	1127
- Error response sent	1128
Individual monitoring event subscriptions delete	
- Request received	1131
- Successful response sent	1132
- Error response sent	1133
Monitoring event notification	
- Request sent	1136
- Successful response received	1137
- Error response received	1138

Table A-3 Device Triggering Statistics

Statistic Name	Statistic ID
Device Triggering transaction	
- Request received	1081
- Successful response sent	1082
- Error response sent	1083
Device Triggering transaction	
- Request received	1086
- Successful response sent	1087
- Error response sent	1088
Individual Device Triggering transaction	
- Request received	1091

Table A-3 (Cont.) Device Triggering Statistics

Statistic Name	Statistic ID
- Successful response sent	1092
- Error response sent	1093
Individual Device Triggering transaction	
- Request received	1096
- Successful response sent	1097
- Error response sent	1098
Individual Device Triggering transaction	
- Request received	1101
- Successful response sent	1102
- Error response sent	1103
Device Triggering delivery report notification	
- Request received	1106
- Successful response sent	1107
- Error response sent	1108

Table A-4 Enhanced Coverage Restriction Statistics

Statistic Name	Statistic ID
Configure	
- Request received	1141
- Successful response sent	1142
- Error response sent	1143
Query	
- Request received	1146
- Successful response sent	1147
- Error response sent	1148

Statistics are stored in the database at a configurable interval, which is configurable using the Admin console. Statistics are collected for the configured interval and stored in database in different records as one per Application per AppServer.

To change the statistics storage interval, follow this procedure:

1. Access the DSR API GW Admin console using https://<Admin-Server-XMI-IP>:9002/console.
2. Login using the admin account created when configuring the API GW.

   The default username is weblogic.
3. Navigate to **OCSG**, and then **AppServerx**, and then **Container Services**, and then **Statistics Service**, and then **Attributes (tab)**.
4. Change the *StoreInterval* to desired seconds.
5. Mark the associated checkbox and click **Update Attributes**.

IPDD Statistics

AAA Messages

Table A-5 AAA Messages

Statistics Name	Description	Statistics ID
TRANSACTION_TYPE_AAA_ACR_REQUEST_SUCCESS	ACR Request Success	2101
TRANSACTION_TYPE_AAA_ACR_REQUEST_FAILURE	ACR Request Failure	2102
TRANSACTION_TYPE_AAA_ACR_START_SUCCESS	ACR Start Success	2111
TRANSACTION_TYPE_AAA_ACR_START_FAILURE	ACR Start Failure	2112
TRANSACTION_TYPE_AAA_ACR_STOP_SUCCESS	ACR Stop Success	2121
TRANSACTION_TYPE_AAA_ACR_STOP_FAILURE	ACR Stop Failure	2122

MQTT Messages

Table A-6 MQTT Messages

Statistics Name	Description	Statistics ID
TRANSACTION_TYPE_MQTT_APP_SUBSCRIPTION_POST_REQ	MQTT App Subscription Post Request	2001
TRANSACTION_TYPE_MQTT_APP_SUBSCRIPTION_POST_RES_SUCCESS	MQTT App Subscription Post Request Success	2002
TRANSACTION_TYPE_MQTT_APP_SUBSCRIPTION_POST_RES_FAILURE	MQTT App Subscription Post Request Failure	2003
TRANSACTION_TYPE_MQTT_APP_SUBSCRIPTION_DELETE_REQ	MQTT App Subscription Delete Request	2011
TRANSACTION_TYPE_MQTT_APP_SUBSCRIPTION_DELETE_RES_SUCCESS	MQTT App Subscription Delete Request Success	2012
TRANSACTION_TYPE_MQTT_APP_SUBSCRIPTION_DELETE_RES_FAILURE	MQTT App Subscription Delete Request Failure	2013
TRANSACTION_TYPE_MQTT_APP_MT_PUBLISH_SINGLE_REQ	MQTT AppMT Publish Single Request	2021
TRANSACTION_TYPE_MQTT_APP_MT_PUBLISH_SINGLE_RES_SUCCESS	MQTT App MT Publish Single Request Success	2022
TRANSACTION_TYPE_MQTT_APP_MT_PUBLISH_SINGLE_RES_FAILURE	MQTT App MT Publish Single Request Failure	2023
TRANSACTION_TYPE_MQTT_APP_MT_PUBLISH_BROAD_CAST	MQTT App MT Publish Broadcast	2031
TRANSACTION_TYPE_MQTT_DEVICE_CONNECT_REQUEST	MQTT Device Connect Request	2041
TRANSACTION_TYPE_MQTT_DEVICE_CONNECT_SUCCESS	MQTT Device Connect Request Success	2042
TRANSACTION_TYPE_MQTT_DEVICE_CONNECT_FAILURE	MQTT Device Connect Request Failure	2043

Table A-6 (Cont.) MQTT Messages

Statistics Name	Description	Statistics ID
TRANSACTION_TYPE_MQTT_DEVICE_SUBSCRIBE_REQUEST	MQTT Device Subscribe Request	2051
TRANSACTION_TYPE_MQTT_DEVICE_SUBSCRIBE_SUCCESS	MQTT Device Subscribe Success	2052
TRANSACTION_TYPE_MQTT_DEVICE_SUBSCRIBE_FAILURE	MQTT Device Subscribe Failure	2053
TRANSACTION_TYPE_MQTT_DEVICE_UNSUBSCRIBE_REQUEST	MQTT Device UnSubscribe Request	2061
TRANSACTION_TYPE_MQTT_DEVICE_UNSUBSCRIBE_SUCCESS	MQTT Device UnSubscribe Success	2062
TRANSACTION_TYPE_MQTT_DEVICE_UNSUBSCRIBE_FAILURE	MQTT Device UnSubscribe Failure	2063
TRANSACTION_TYPE_MQTT_DEVICE_PUBLISH_REQUEST	MQTT Device Publish Request	2071
TRANSACTION_TYPE_MQTT_DEVICE_PUBLISH_SUCCESS	MQTT Device Publish Success	2072
TRANSACTION_TYPE_MQTT_DEVICE_PUBLISH_FAILURE	MQTT Device Publish Failure	2073
TRANSACTION_TYPE_MQTT_DEVICE_DISCONNECT_REQUEST	MQTT Device Disconnect Request	2081
TRANSACTION_TYPE_MQTT_DEVICE_PUBLISH_NOTIFY	MQTT Device Publish Notify	2091
TRANSACTION_TYPE_MQTT_DEVICE_PUBLISH_NOTIFY_SUCCESS	MQTT Device Publish Notify Success	2092
TRANSACTION_TYPE_MQTT_DEVICE_PUBLISH_NOTIFY_FAILURE	MQTT Device Publish Notify Failure	2093

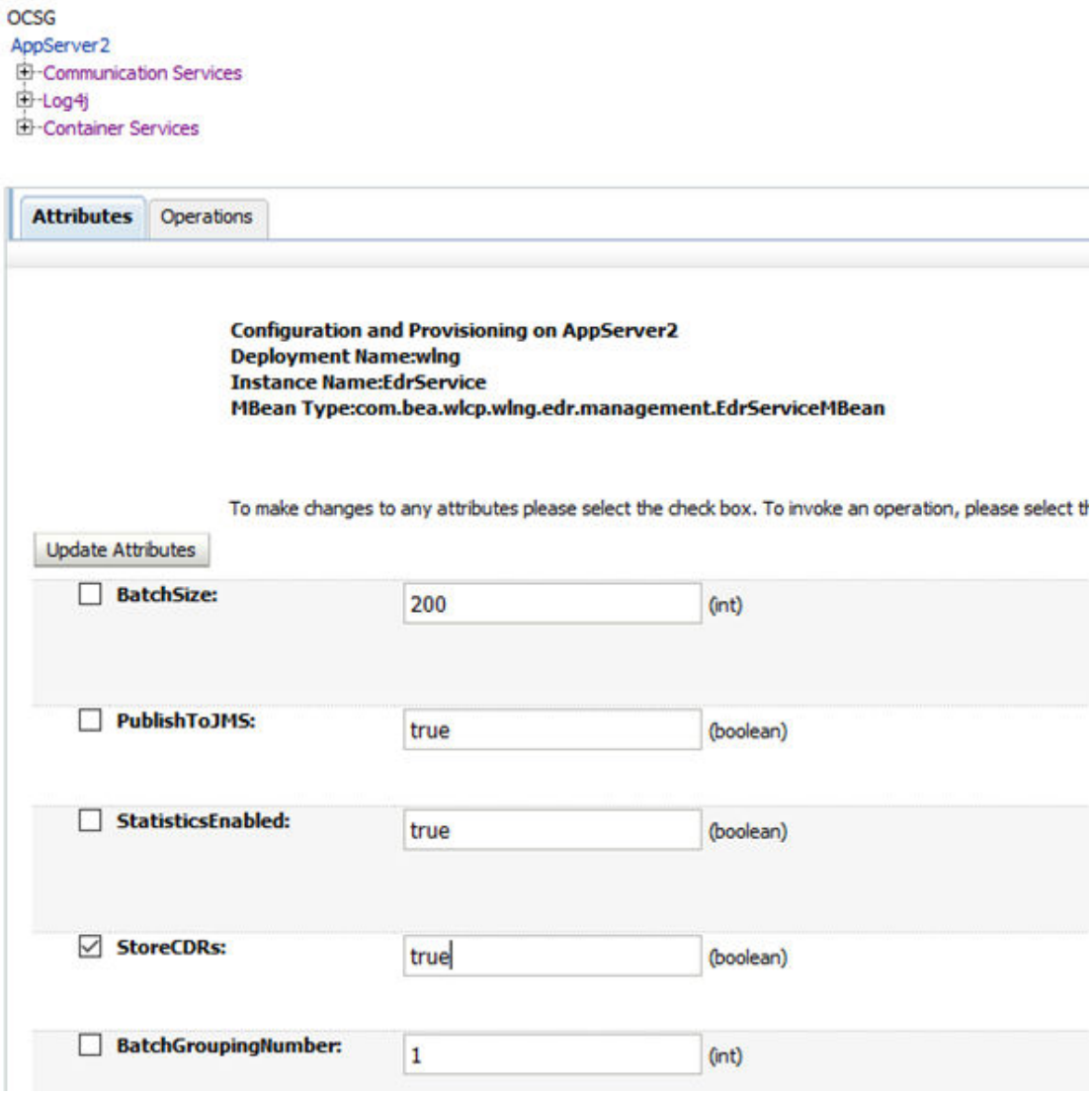
Enable CDRs

Any change made to an Application Server (AppServer) reflects to all other servers.

1. Access the DSR API GW Admin console using https://<Admin-Server-XMI-IP>:9002/console.
2. Login using the admin account created when configuring the API GW.

   The default username is weblogic.
3. Navigate to **OCSG**, and then **AppServerx**, and then **Container Services**, and then **Edr Service** , and then **Attributes (tab)**.
4. Change the *StoreCdrs* parameter to true.
5. Mark the associated checkbox and click **Update Attributes**.

Figure A-7 Enable CDRs



Start/Restart Administrative Server

The procedure to start and restart the server is the same.

1. Make sure the nodemgr service is running on the Admin server by with this command:

   ```
   sudo service nodemgr status
   ```

   If the nodemgr service is not running, start the service:

   ```
   sudo service nodemgr start
   ```

2. cd to /u03/app/oracle/ocsg-x.x.x/user_projects/domains/services-gatekeeper-domain.

3. Execute

   ```
   source ../../../wlserver/server/bin/setWLSEnv.sh
   ```

4. Execute

   ```
   java weblogic.WLST", this will start a WLST prompt
   ```

5. Execute this command at the WLDT prompt

   ```
   Wls:/offiline: nmConnect('<nodemanager-user-name', 'nodemanager-password', 'adminserver-imi-ip', '5556', 'services-gatekeeper-domain', '/u03/app/oracle/ocsg-x.x.x/user_projects/domains/services-gatekeeper-domain','plain')
   ```

   nodemanager-user-name - user name of node manager provided while configuring DSR API GW

   nodemanager-password - password of node manager provided while configuring DSR AI GW

   ocsg-x.x.x. - current DSR API GW version installed

6. Execute

   ```
   wls:/nm/services-gatekeeper-domain> nmStart('AdminServer')
   ```

7. Make sure the Admin server has successfully started by accessing the console URL at https://<Admin-Server-XMI-IP>:9002/console.

Start/Restart Application Server

The procedure to start and restart the server is the same.

1. Make sure the nodemgr service is running on the Admin server by with this command:

   ```
   sudo service nodemgr status
   ```

   If the nodemgr service is not running, start the service:

   ```
   sudo service nodemgr start
   ```

2. Access the DSR API GW Admin console using https://<Admin-Server-XMI-IP>:9002/console.
3. Login using the admin account created when configuring the API GW.

   The default username is weblogic.

4. Navigate to **Environment**, and then **Servers**, and then **Control (tab)**.
5. Click **Start** and confirm.

Stop the Administrative and Application Servers

1. Access the DSR API GW Admin console using https://<Admin-Server-XMI-IP>:9002/console.
2. Login using the admin account created when configuring the API GW.

   The default username is weblogic.
3. Navigate to **Environment**, and then **Servers**, and then **Control (tab)**.
4. Select the server to stop and click **Stop**.
5. Select **Force shutdown now** and confirm.

Alarms

Alarms are raised by DSR API GW for different events. If SNMP is configured, alarms are sent as SNMP traps. OCSG alarms related to SCEF are described in the Alarms and KPIs reference guide.

Add New XSI to OCSG

This procedure adds a new External Signaling Interface (XSI) to the Oracle Communications Services Gatekeeper (OCSG).

1. Attach XSI interface to VMs.
2. Configure ifcfg files so the network is configured and the IP address is picked up by the VM.
3. Run these command to open ports in the firewall for the new XSI:

```
sudo iptablesAdm append
                  --type=rule --protocol=IPv4 --domain=01dsrapigw --
table=filter --chain=INPUT
                  --match='-m state --state NEW -m tcp -p tcp --dport
10001 -d <XSI-IP> -j
                  ACCEPT' --persist=yes


sudo iptablesAdm append
                  --type=rule --protocol=IPv4 --domain=01dsrapigw --
table=filter --chain=INPUT
                  --match='-m state --state NEW -m tcp -p tcp --dport
10002 -d <XSI-IP> -j
                  ACCEPT' --persist=yes
```

4. Access the DSR API GW Admin console using https://<Admin-Server-XMI-IP>:9002/console.
5. Login using the admin account created when configuring the API GW.

   The default username is weblogic.
6. Navigate to **Environment**, and then **Servers**, and then **AppServerx**, and then **Protocols**, and then **Channels**.
7. Click **Lock & Edit**.

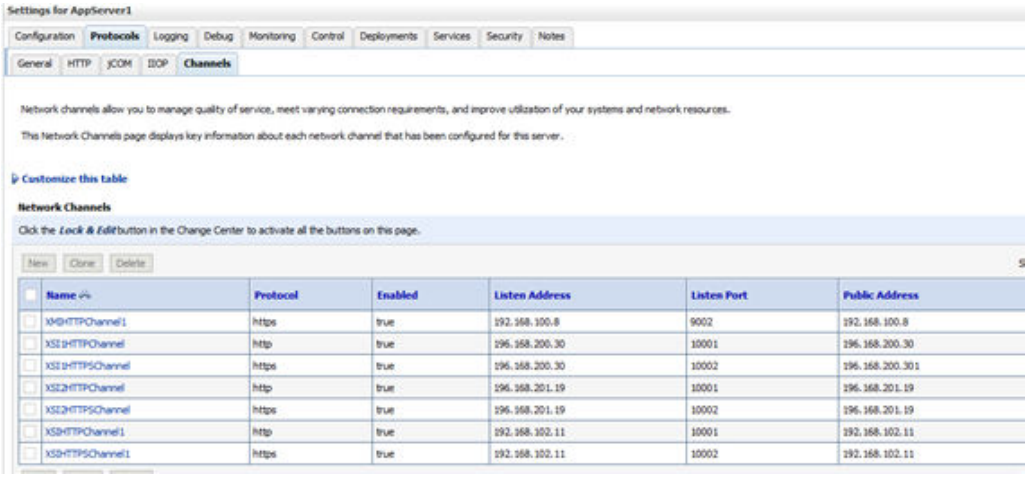
8. Add new channels.

   Each channel name should be unique.

   Change the name, protocol, IP and port.

   Leave the rest of the options as default.
9. Add a channel for the new XSI IP and 10001 for http protocol.
10. Add another channel for XSI and 10002 for https protocol.

Figure A-8 Add New XSI to OCSG



Change the Administrative Console Account Password

1. Access the DSR API GW Admin console using https://<Admin-Server-XMI-IP>:9002/console.
2. Login using the admin account created when configuring the API GW.

   The default username is weblogic.
3. Navigate to **Security Realms**, and then **myrealm**, and then **User and Groups**, and then **Users** , and then **weblogic**, and then **passwords**.
4. Stop all the Administrative and Application servers.

   See the Stop the Administrative and Application Servers procedure.
5. In each server, navigate to the u03/app/oracle/ocsg-x.x.x/user_projects/domains/services-gatekeeper-domain/servers/<Server-name>/security folder.
6. Delete the boot.properties file.
7. Recreate the boot.properties file with a new username and password.

   username=<user-name>

   Password=<password>

> **Note:**
>
> These details are encrypted when the server starts successfully.

8. Stare all the Administrative and Application servers.

   See the Start/Restart Administrative Server and Start/Restart Application Server procedures.

Create User Account

1. Access the DSR API GW Admin console using https://<Admin-Server-XMI-IP>:9002/console.
2. Login using the admin account created when configuring the API GW.

   The default username is weblogic.
3. Navigate to **OCSG**, and then **AppServerx**, and then **Container Services**, and then **Management Users** , and then **ManagementUsers**, and then **Operations (tab)**.
4. Select *addUser*.
5. Provide the new username, password, Userlevel (1000), and Type (1) and click **Invoke**.

Change the Operator Account Password

1. Access the DSR API GW Admin console using https://<Admin-Server-XMI-IP>:9002/console.
2. Login using the admin account created when configuring the API GW.

   The default username is weblogic.
3. Navigate to **OCSG**, and then **AppServerx**, and then **Container Services**, and then **Management Users** , and then **ManagementUsers**, and then **Operations (tab)**.
4. Select *setUserPassword*.
5. Provide the new username and password and click **Submit**.

Purge Database Tables

This section lists the database tables you should purge periodically on the OCSG database. How often the tables are cleaned depends on the traffic capacity of the site. This section provides some recommendations.

About Cleaning Database Tables

Table A-7 lists the database tables that you must periodically clean to prevent them from growing too large and adversely affecting performance.

Table A-7 Database Table Cleaning Intervals

Table	Recommended Cleaning Interval
SLEE_ALARM	Every two months
SLEE_CHARGING (if cdrs is enabled)	Depends on the traffic capacity of the site

Table A-7 (Cont.) Database Table Cleaning Intervals

Table	Recommended Cleaning Interval
SLEE_STATISTICS_DATA	Every month

Set Up Two-Way SSL Configuration

Two-way SSL configuration mandates clients, opening an HTTPS connection to DSR API GW, present a client certificate (that is validated by DSR API GW) before opening a connection.

The trusted client certificate should be imported to DSR API GW server so the validation is successful.

Import Client Certificate

This procedure to import the client certificate is performed on each AppServer.

1. Access the DSR API GW Admin console using https://<Admin-Server-XMI-IP>:9002/console.
2. Login using the admin account created when configuring the API GW.

   The default username is weblogic.
3. Navigate to **Environment**, and then **Servers**, and then **AdminServer or AppServerx**, and then **Configuration**, and then **KeyStore**.
4. Note the trust store file path.
5. SSH to the corresponding server and browse to the trust store file path.
6. Copy the client certificate(.cer file) to import to the current directory.
7. Execute this command to import the certificate to the trust store (trust store passphrase should be entered):

   ```
   keytool -import -alias <any-alias-name-for-cert> -file <certificate-file> -keystore <trust-store-name>
   ```
8. Restart the corresponding server.
9. Access the DSR API GW Admin console using https://<Admin-Server-XMI-IP>:9002/console.
10. Login using the admin account created when configuring the API GW.

    The default username is weblogic.
11. Navigate to **Environment**, and then **Servers**, and then **AdminServer or AppServerx**, and then **Configuration**, and then **SSL**.
12. Click **Advanced**.
13. Click **Lock and Edit**.
14. Change *Two Way Client Cert Behavior* to `Client Certs Requested And Enforced`.
15. Click **Save** and **Active Changes**.

Import Server Certificate

This procedure imports the server certificates on the application servers, which the DSR API GW then sends to SCEF reports. The request is initiated by the DSR API GW towards the application server over HTTPS.

1. Access the DSR API GW Admin console using https://<Admin-Server-XMI-IP>:9002/console.
2. Login using the admin account created when configuring the API GW.

   The default username is weblogic.
3. Navigate to **Environment**, and then **Servers**, and then **AdminServer or AppServerx**, and then **Configuration**, and then **KeyStore**.
4. Note the trust store file path.
5. SSH to the corresponding server and browse to the trust store file path.
6. Copy the client certificate(.cer file) to import to the current directory.
7. Execute this command to import the certificate to the trust store (trust store passphrase should be entered):

   ```
   keytool -import -alias <any-alias-name-for-cert> -file <certificate-file>
   -keystore <trust-store-name>
   ```
8. Restart the corresponding server.

Change SSL Certificates and Private Keys

DSR API GW shipped with a demo certificate and private key, which are not recommended for use in a production environment.

To change the demo certificate and private keys of DSR API GW, obtain:

- A CA signed digital certificate (.pem file) and private key for each DSR API GW server separately.
- A root certificate of CA and any other intermediate certificates used to sign the digital certificate.

This procedure is performed on each AppServer.

1. SSH to the server.
2. Browse to the /u03/app/oracle/ocsg-x.x.x/user_projects/domains/ services-gatekeeper-domain/security directory.

   Replace x.x.x with the DSR API GW version.
3. Copy the signed certificate, private key, CA root and intermediate certificates (if any) to the current directory.
4. Execute

   ```
   source ../../../../wlserver/server/bin/setWLSEnv.sh
   ```

5. Create a custom key store and import the private key and signed digital certificate with this command.

```
java utils.ImportPrivateKey -keystore  SeverIdentity.jks -storepass
<storepass>
 -storetype JKS -keypass <keypass> -alias <skey> -certfile
<serverCert.pem> -keyfile <ServerKey.pem>
 -keyfilepass <keypass>
```

   Keystore: SeverIdentity.jks -JKS file in which the certificate and key will be imported.

   Storepass: storepass - This is the password of the keystore file severIdentity.jks

   Storetype: JKS - Java Key Store.

   Keypass: keypass - This password will be configured in server which will be used to read the Private Key from the keystore.

   Alias: skey - This is the alias used for reading the Private Key from the Keystore.

   Certfile: serverCert.pem - This is the certificate to be imported into the Keystore.

   Keyfile: ServerKey.pem - This is the Private Key to be imported into the Keystore.

   Keyfilepass: keypass - This is the Password required to read the Private Key from the ServerKey.pem file

6. Create a custom trust store (java key store) and import the CA root certificate with this command.

```
keytool -import -file <ca.cert> -alias <firstCA> -keystore
<ServerTrust.jks> -storepass <storepass>
```

   ca.cert - ca root certificate to be imported

   firstCA - an alias to the certificate

   ServerTrust.jks - trust store with the name will be created

   Storepass - trust store pass phrase

7. Access the DSR API GW Admin console using https://<Admin-Server-XMI-IP>:9002/console.
8. Login using the admin account created when configuring the API GW.

   The default username is weblogic.

9. Navigate to **Environment**, and then **Servers**, and then **AdminServer or AppServerx**, and then **Configuration**, and then **KeyStore**.
10. Click **Advanced**.
11. Click **Lock and Edit**.
12. Change *Keystores* to `Custom Identity and Custom Trust`.
13. Provide these values:
    - Custom Identity Keystore
    - Custom Identity Keystore Type
    - Custom Identity Keystore Passphrase

- Confirm Custom Identity Keystore Passphrase
- Custom Trust Keystore
- Custom Trust Keystore Type
- Custom Trust Keystore Passphrase
- Confirm Custom Trust Keystore Passphrase

14. Select the SSL tab and provide these values:
    - Private Key Alias
    - Private Key Passphrase
    - Confirm Private Key Passphrase
15. Click **Activate Changes**.
16. Navigate to **Environment**, and then **Servers**, and then **Control (tab)** to restart the server.

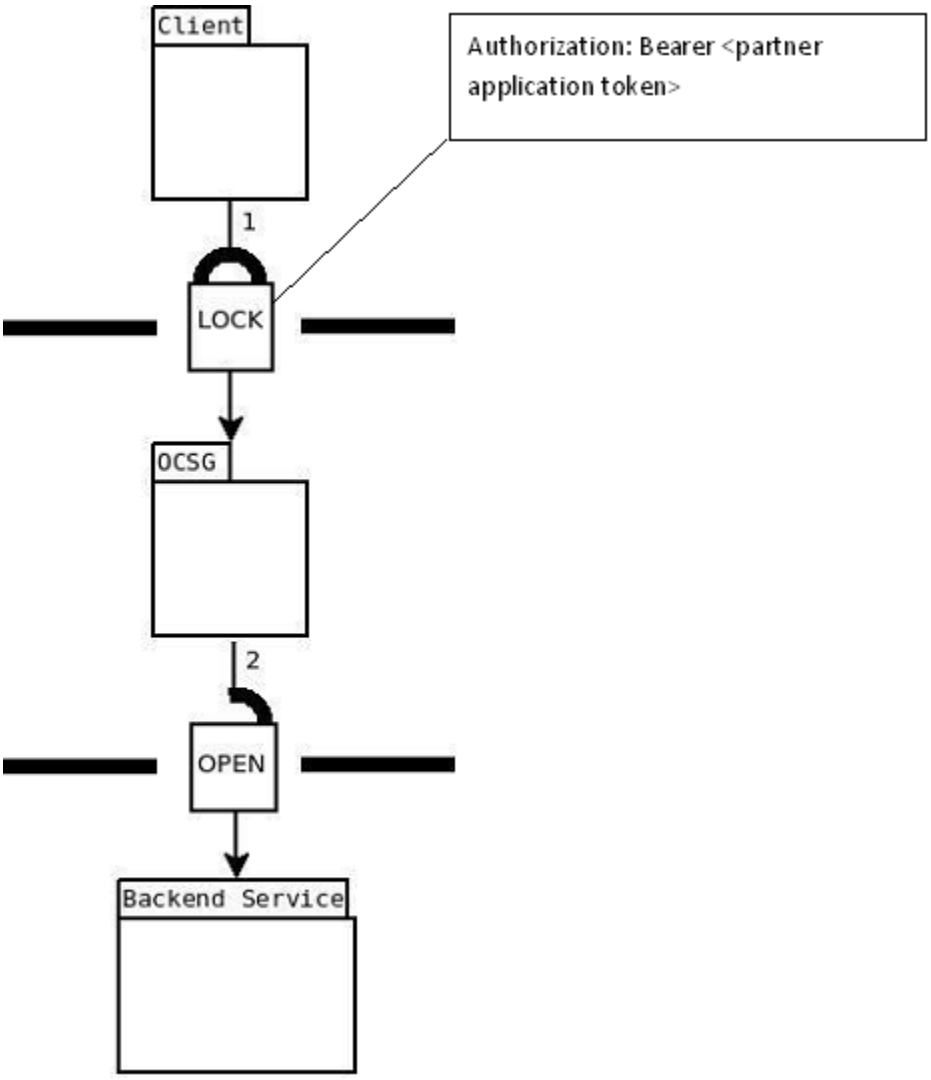
Open Authorization Configuration Overview

Open Authorization or OAuth is an open standard for token-based authentication and authorization on the Internet. OAuth allows an end user's account information to be used by third-party services, such as Facebook, without exposing the user's password. This section describes an alternative configuration to modifying the APIs to authenticate with OCSG. The installation script automatically creates the APIs with support for OAuth as shown in Figure A-9.

Figure A-9 OAuth Installation Script

Authorization take place after client has been created and between the two firewalls as shown in Figure A-10.

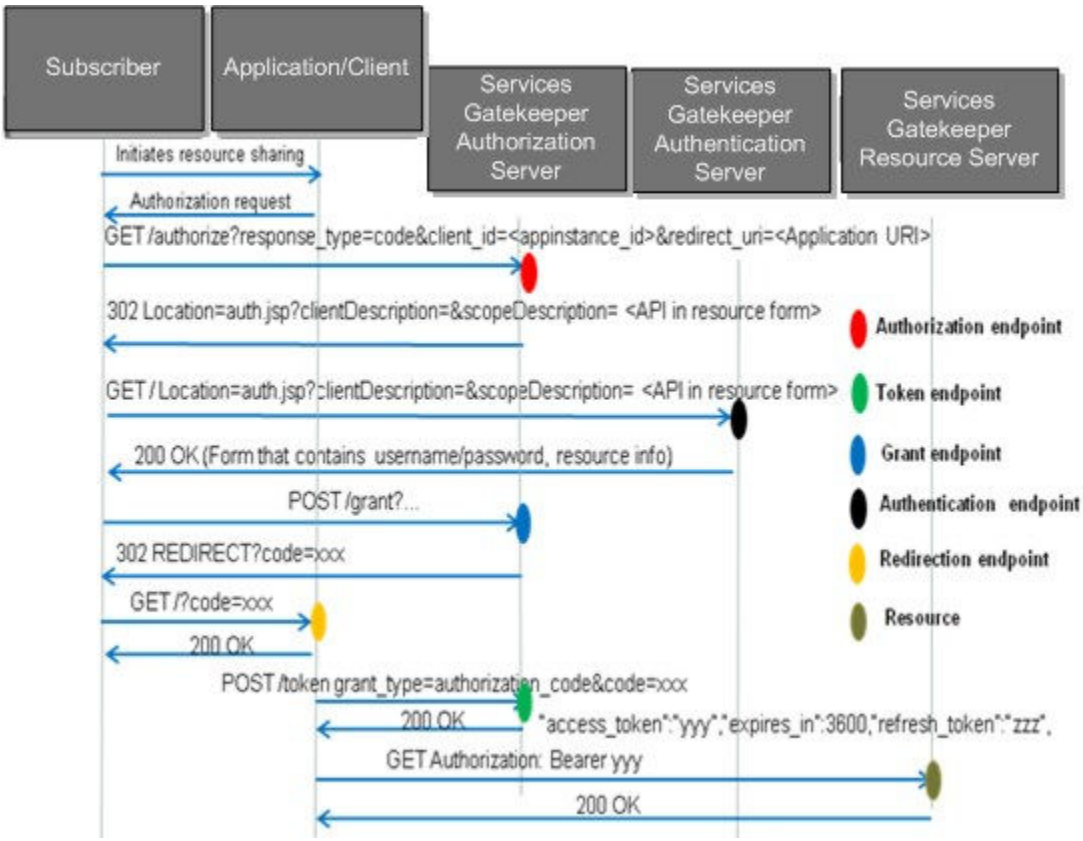
Figure A-10 Authorization Overview



This section assumes an API has been created and published and that the corresponding partner application has also been created. After the application has been created, assigned to a group, set up with the user account, set up the authorization as described in this section.

The Authorization Code grant type is used by confidential and public clients to exchange an authorization code for an access token. After the user returns to the client via the redirect URL, the application acquires the authorization code from the URL and uses it to request an access token. Figure A-11 shows this process using the resource owner authentication and code grant redirect.

Figure A-11 OAuth Code Grant



Set Up Authentication and Grant Redirect URLs

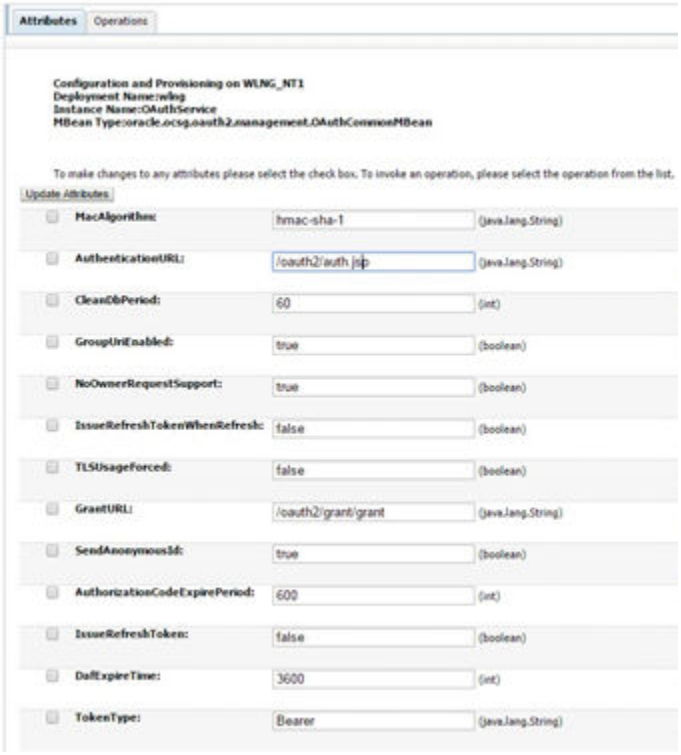
The first step to authorization is to set up the authentication and grant redirect URLs.

1. Navigate to **OCSG**, and then **AppServerx**, and then **Container Services**, and then **OAuthService**, and then **OAuthCommonMBean**.

Figure A-12 OAuthCommonMBean

2. Set up the authentication and grand redirect URLs.

Figure A-13 Authentication and Grand Redirect URLs

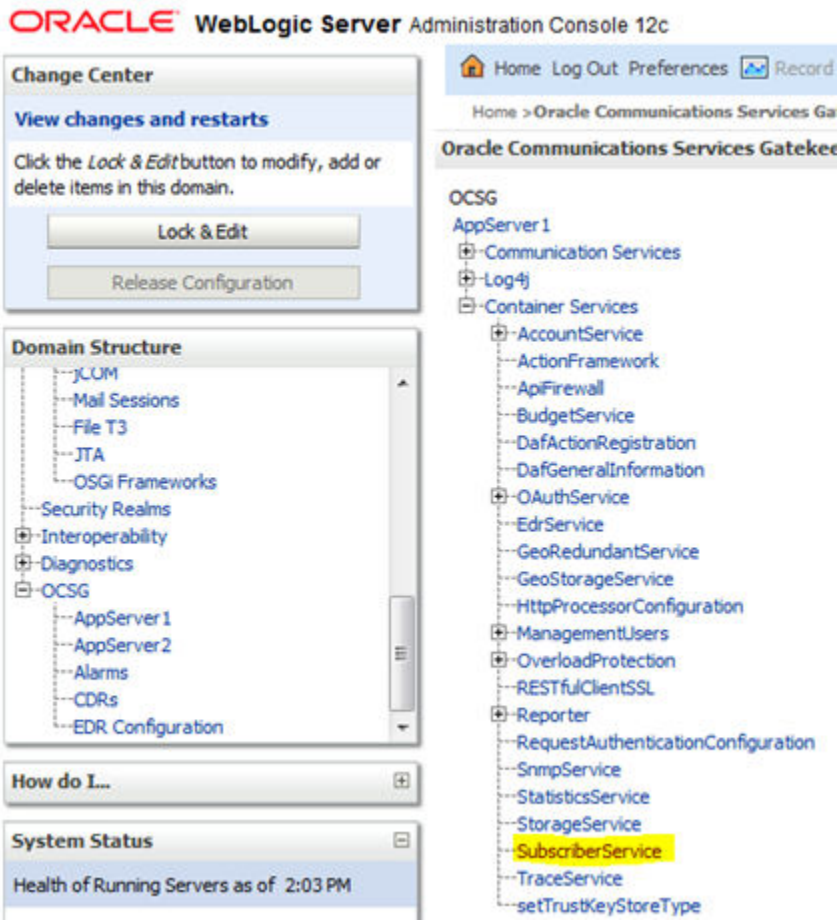


Subscriber

The Services Gatekeeper offers a built-in subscriber repository to authenticate subscribers. To use the default Subscriber Manager to authenticate subscribers, follow these steps:

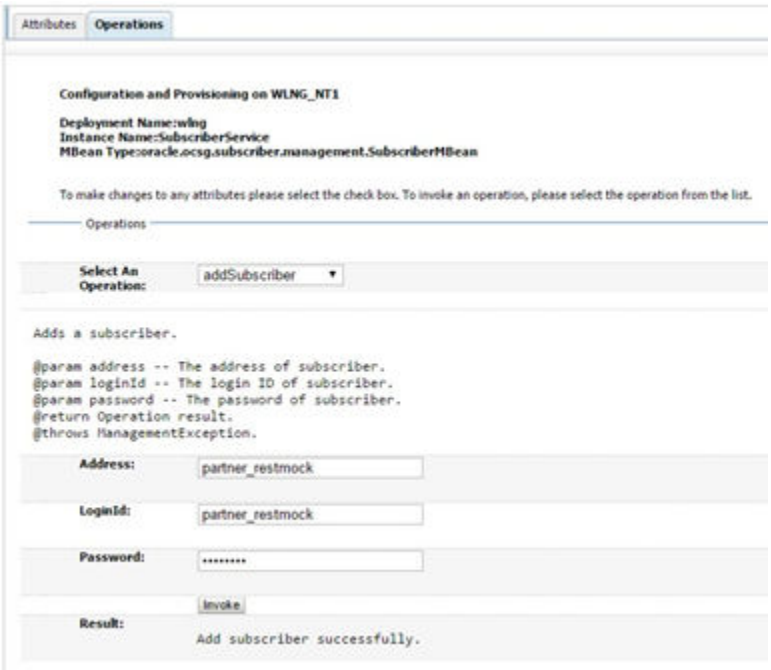
1. Navigate to **OCSG**, and then **AppServerx**, and then **Container Services**, and then **SubscriberService**.

Figure A-14 SubscriberService



2. Create a subscriber account to use for authentication purposes.

Figure A-15 Subscriber

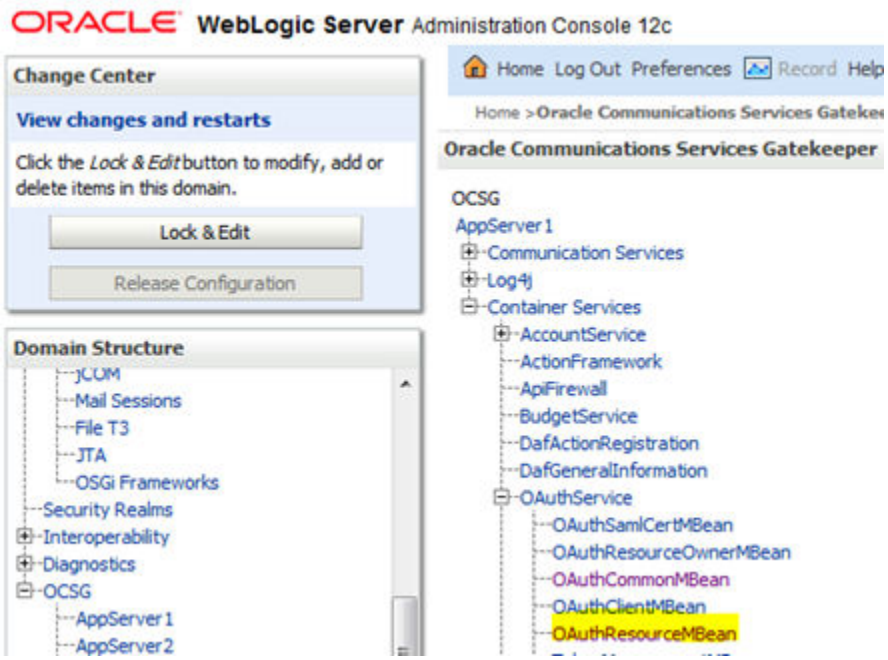


Resource Owner

The final step to open authorization is to set up the resource owner and associated resources.

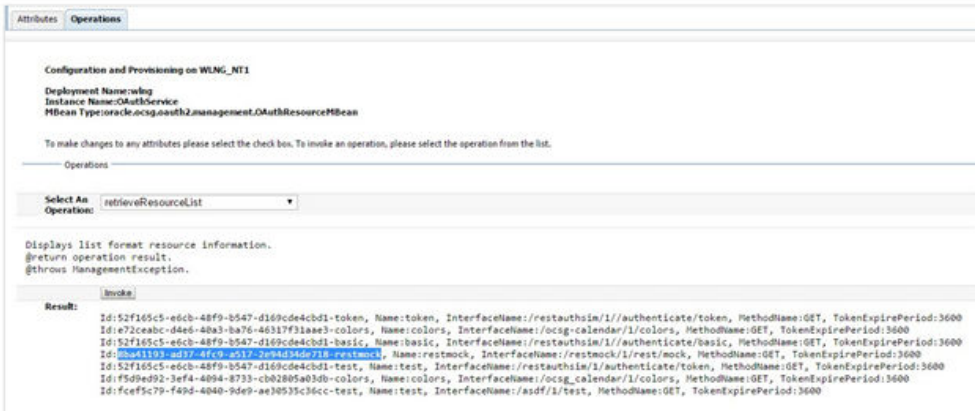
1. Find the resource ID by navigating to **OCSG**, and then **AppServerx**, and then **Container Services**, and then **OAuthService**, and then **OAuthResourceMBean**.

Figure A-16 OAuthResourceMBean



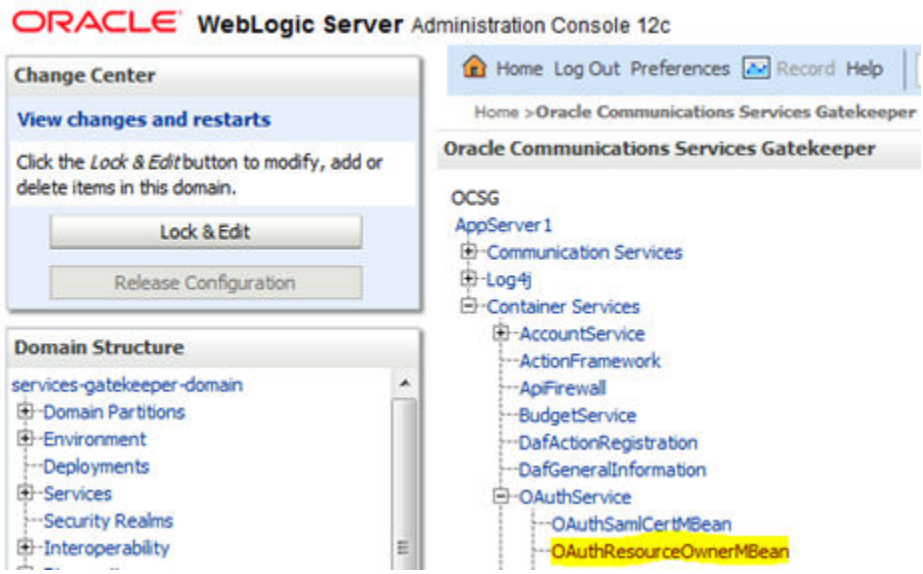
2. Find the resource ID corresponding to the application to be authenticated.

Figure A-17 Resource ID



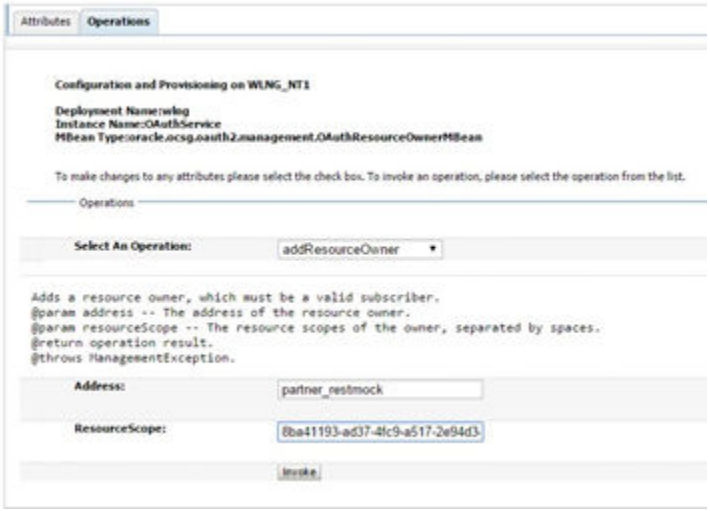
3. Navigate to **OCSG**, and then **AppServerx**, and then **Container Services**, and then **OAuthService**, and then **OAuthResourceOwnerMBean**..

Figure A-18 OAuthResourceOwnerMBean



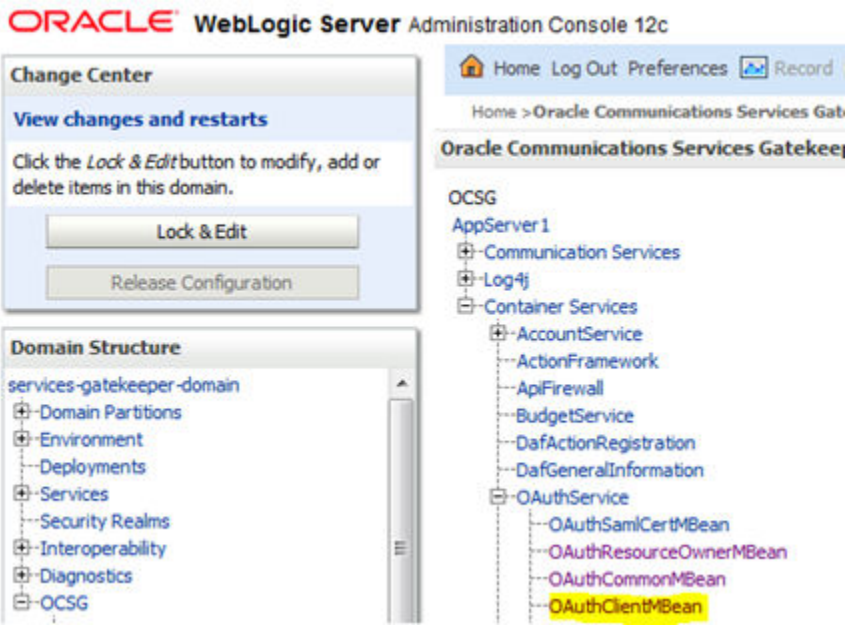
4. Add the subscriber as resource owner of the application ID previously identified.

Figure A-19 Add Subscriber as Resource Owner



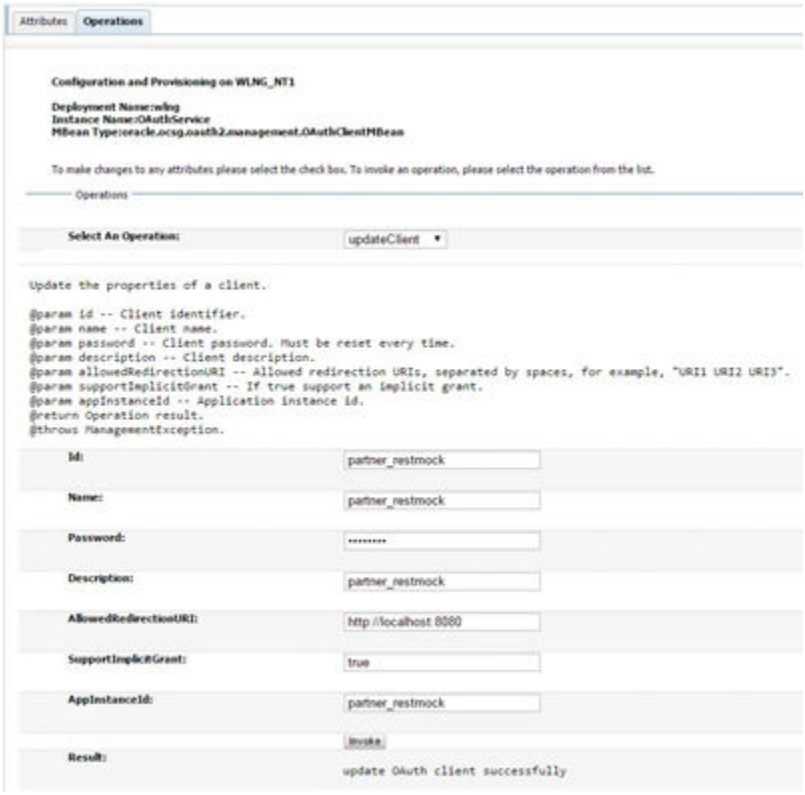
5. Navigate to **OCSG**, and then **AppServerx**, and then **OAuthService**, and then **OAuthClientMBean**.

Figure A-20 OAuthClientMBean



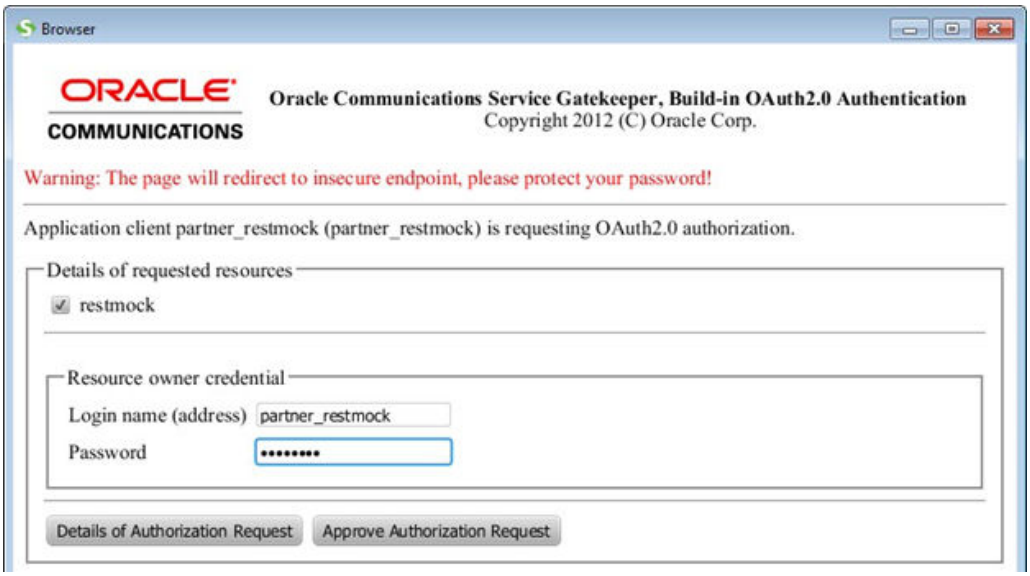
6. Add the application traffic user as a client allowed redirect.

Figure A-21 Traffic User



Now when trying to access the application, this screen displays to request authentication.

Figure A-22 Authentication Request

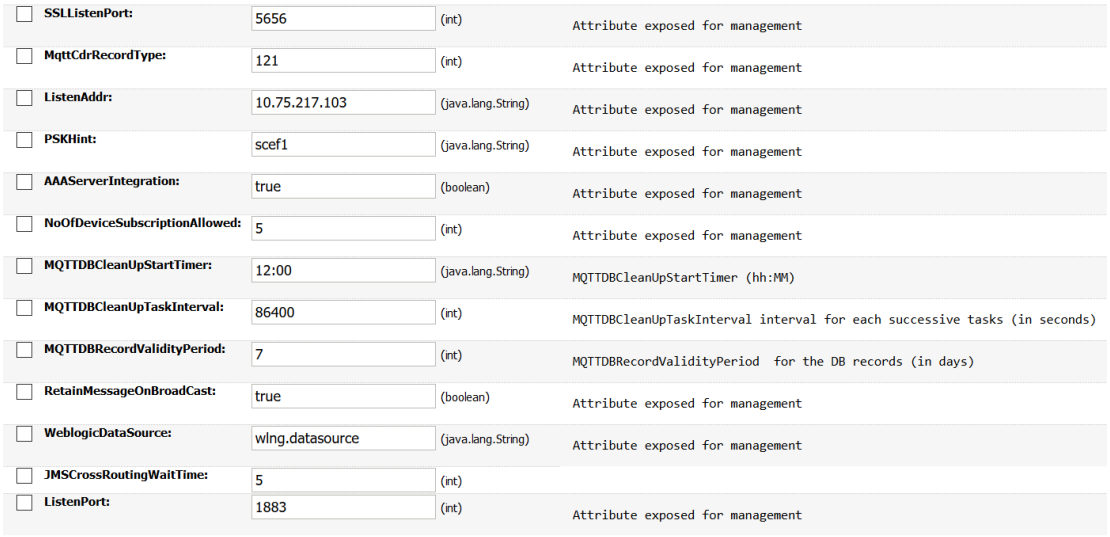


MQTT Configuration

MQTT Broker exposes Management bean, accessible via OCSG Admin Portal.

1. Access the DSR API GW Admin console using `https://<Admin-Server-XMI-IP>:9002/console`.
2. Login using the admin account created when configuring the APIGW.
3. Navigate to **OCSG**, and then **AppServerx**, and then **Communication Services**, and then **Mqttbroker_rest_plugin_Instancexxx**.
4. Enter the required values for the fields shown in Figure A-23.

Figure A-23 MQTT Configuration



Attributes:

`ListenPort`

Description: TCP Listen port of MQTT Broker.

Scope: Local to AppServer.

`AAAServerIntegration`

Description: Enable/Disable AAA Integration. Option enabled always. Option given for future enhancements.

Scope: Shared across AppServers. Changes on one AppServer reflects on all other AppServers.

`PSKHint`

Description: PSK hint presented by MQTT broker in a response to MQTT Device TCP hello request with PSK cipher support.

Scope: Cluster, Shared across AppServers. Changes on one AppServer reflects on all other AppServers.

`NoOfDeviceSubscriptionAllowed`

Description: Limit on number of subscriptions received per device.

Range: 1-10

Scope: Cluster, Shared across AppServers. Changes on one AppServer reflects on all other AppServers.

`WeblogicDataSource`

Description: Data source name configured on OCSG, used to connect to MQTT database of broker. By default, it is `wlng.datasource`. This attribute changes only in case MQTT broker point to database other than OCSG DB.

Scope: Cluster, Shared across AppServers. Changes on one AppServer reflects on all other AppServers.

`JMSCrossRoutingWaitTime`

Description: Application T8 requests are cross-routed across AppServers when the MQTT device connects to different AppServer than the one received in T8 request. This parameter defines the waiting time, the originating AppServer waits for a response for a cross-routed message. This parameter is applicable only in case of MT message to a single device.

Range: 1-10 (in seconds)

Scope: Cluster, Shared across AppServers. Changes on one AppServer reflects on all other AppServers.

`SSLListenPort`

Description: SSL Listen port of MQTT Broker.

Scope: Local to AppServer.

`ListenAddr`

Description: IP Address of server, MQTT Broker listen for client connections.

Scope: Local to each AppServer.

`MQTTDBCleanUpStartTimer`

Description: `MQTTDBCleanUpStartTimer` (hh:MM)

Scope: Cluster, Shared across AppServers. Changes on one AppServer reflects on all other AppServers.

`MQTTDBCleanUpTaskInterval`

Description : `MQTTDBCleanUpTaskInterval` interval for each successive tasks (in seconds)

Scope: Cluster, Shared across AppServers. Changes on one AppServer reflects on all other AppServers.

`MQTTDBRecordValidityPeriod`

Description: `MQTTDBRecordValidityPeriod` for the DB records (in days)

Scope: Cluster, Shared across AppServers. Changes on one AppServer reflects on all other AppServers.

`RetainMessageOnBroadCast`

Description: This flag will decide if message to be retained on broadcast (true or false)

Scope: Cluster, Shared across AppServers. Changes on one AppServer reflects on all other AppServers.

`MqttCdrRecordType`

Description: MQTT CDR record type

Scope: Cluster, Shared across AppServers. Changes on one AppServer reflects on all other AppServers.

Operations:

`addOrUpdateMQTTAPNRateControl`

Description: This operation adds new APN Rate control or updates existing rate control with given values.

Input:

- **APNName**: Name of the APN, such as, `oracle.com`
- **APNRate**: Rate of the messages allowed from this APN per device. Value should be between 1 - 2, 147, 483, 647.
- **APNPeriod**: Period for which the defined rate is applicable. Unit is seconds. Consumption rate is reset at start of period. Value should be between 1- 2, 147, 483, 647.

Scope: cluster, performing operation on one Appserver is sufficient.

`deleteMQTTAPNRateControl`

Description: This operation deletes the APN rate control set.

Input:

APNName: APN name to be removed.

Scope: cluster, performing operation on one Appserver is sufficient.

`listAllMQTTAPNRateControl`

Description: This operation lists the configured APN rate control sets.

Output: Displays the configured APN rate control sets.

For example:

apn2@oracle.com, 20, 20

apn1@oracle.com, 3, 120

`addOrUpdatePSKInfo`

Description: This operation adds or updates the PSK information for MQTT devices.

Input: Absolute path of `csv` file containing `psk` information as `<mqtt_client_id,psk>`

Note: PSK CSV file shall be copied to one of the AppServer. Invoking operation on AppServer is sufficient to configure the PSK for mqtt devices. PSK is encrypted and stored in database based on OCSG domain security identities. It is important to backup the domain folder in Admin server periodically. For more details, refer to *DSR API Gateway Installation Guide*.

Scope: cluster, performing operation on one Appserver is sufficient.

`deletePSKInfo`

Description: This operation delete the PSK information for MQTT devices.

Input: Absolute path of csv file containing MQTT client id

Note: Copy the csv file to one of the AppServer. Invoking operation on AppServer is sufficient to delete the PSK for MQTT devices.

Scope: cluster, performing operation on one Appserver is sufficient.

`listAllPSKCipherSuites`

Description: This operation lists all PSK ciphers enabled in MQTT Broker.

Output: List of enabled PSK ciphers.

`updateSupportedPSKCipherSuites`

Description: This operation updates the list of PSK ciphers enabled in MQTT Broker.

Input: Comma separated list of PSK ciphers must be enabled.

Note: This operation will cause MQTT broker to restart.

Scope: cluster, performing operation on one Appserver is sufficient.

`listAllMQTTDBCleanUpList`

Description: This operation lists the table names that are part of DB cleanup schedule in MQTT Broker.

Output: List the table names.

`updateMQTTDBCleanUpList`

Description : This operation updates the list of table names (provided with comma separated) to schedule DB cleanup.

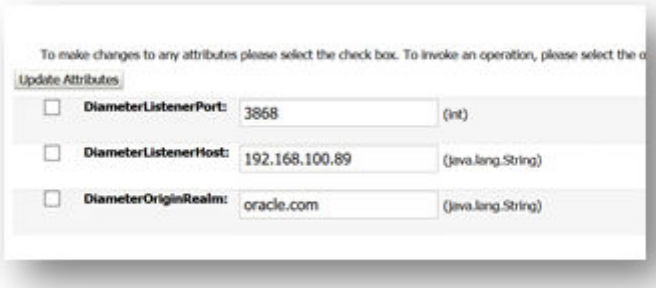
Input: Comma separated list of table names to be scheduled.

SCEF AAA Configuration

A Management bean is exposed to configure diameter configurations.

Navigate to **Service-gatekeeper-domain**, and then **OCSG**, and then **AppServerx**, and then **Communication Services**, and then **SCEF_AAA_Configurationxxxxx**.

Figure A-24 SCEF AAA configuration



Attributes:

`DiameterListernerPort`

Description: Diameter server port.

Scope: Local to Appserver.

`DiameterListenerHost`

Description: Diameter server hostname/IP. This can also be used to configure OCSG AAA FQDN.

Scope: Local to Appserver.

Steps to configure OCSG AAA FQDN:

1. In the `Appserver1`, execute `"sudo vim /etc/hosts"`

   Add the following line at the end, such as:

   ```
   "<IPaddress of Appserver1 on which 3868 port is opened > scefa01-acc-dr.scef.msk.iot.mnc001.mcc250.3gppnetwork.org"
   ```

   For example:

   ```
   200.168.100.3 scefa01-acc-dr.scef.msk.iot.mnc001.mcc250.3gppnetwork.org
   ```

2. Navigate to **Admin WebLogic Console**, and then **Domain Structure**, and then **OCSG**, and then **AppServer1**, and then **Communication Services**, and then **SCEF_AAA_Configuration**

3. Modify Diameter host with `Fqdn` instead of existing IP address.
4. Repeat above steps on all the Appservers.

`DiameterOriginRealm`

Description: Diameter server realm.

Scope:
cluster, shared across Appservers.

QoS Control Configuration

Following needs to be configured for QoS Control:

1. Manage ScsAs QoS Configuration
2. Manage QoS Reference Configuration

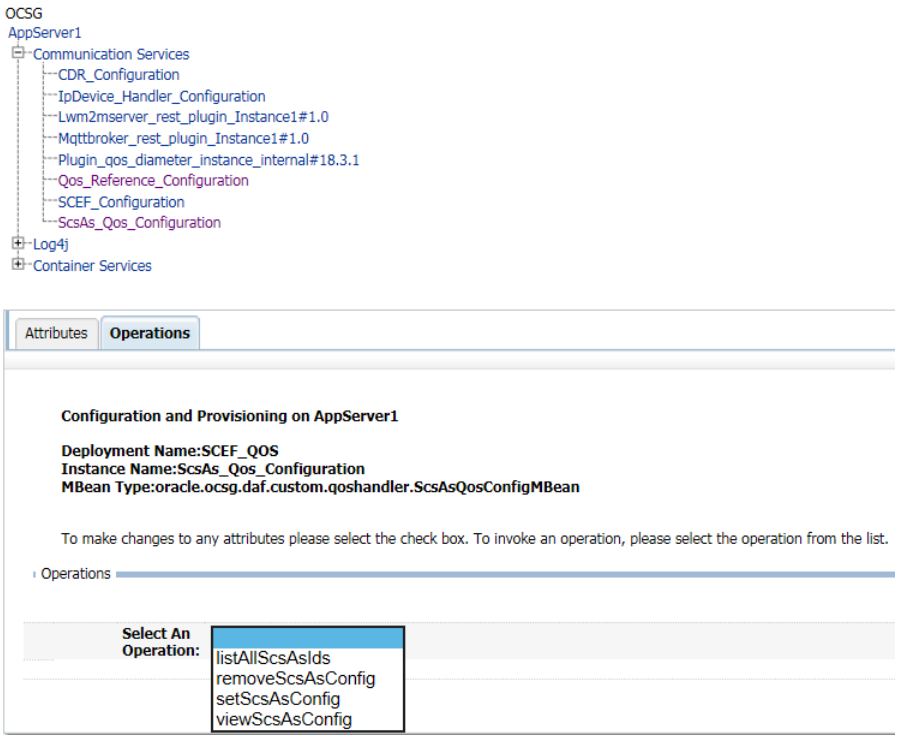
Manage ScsAs QoS Configuration

Currently this configuration allows the user to configure **Qos Reference ID** and **Max AS session duration** for every SCS. Following are the steps to manage this configuration:

> **Note:**
>
> Any change made to an Application Server (AppServer) reflects to all other servers.

1. Access the DSR API GW Admin console using https://<Admin-Server-XMI-IP>:9002/console
2. Login using the admin account created while configuring the API GW. The default username is **weblogic**
3. Navigate to ***OCSG > AppServerx > Communication Services > ScsAs_Qos_Configuration > Operations*** (tab)

Figure A-25 Manage ScsAs QoS Configuration



Add/Modify ScsAs QoS Configuration

Follow this procedure to add/modify ScsAs:

1. Select **setScsAsConfig**.
2. Enter the **ScsAs ID** of the SCS, **default QosReference value**, and **max AS session duration** for the SCS.
3. Click **Invoke**.

View ScsAs Qos Configuration

Follow this procedure to view ScsAs:

1. Select **viewScsAsConfig**.
2. Enter the **ScsAs ID** of the SCS and click **Invoke**.

Remove a ScsAs Qos Configuration

Follow this procedure to remove ScsAs:

1. Select **removeScsAsConfig**.
2. Enter the **ScsAs ID** of the SCS and click **Invoke**.

List All ScsAs IDs Configured

This procedure lists all ScsAs IDs:

Select **listAllScsAsIds** and click **Invoke**.

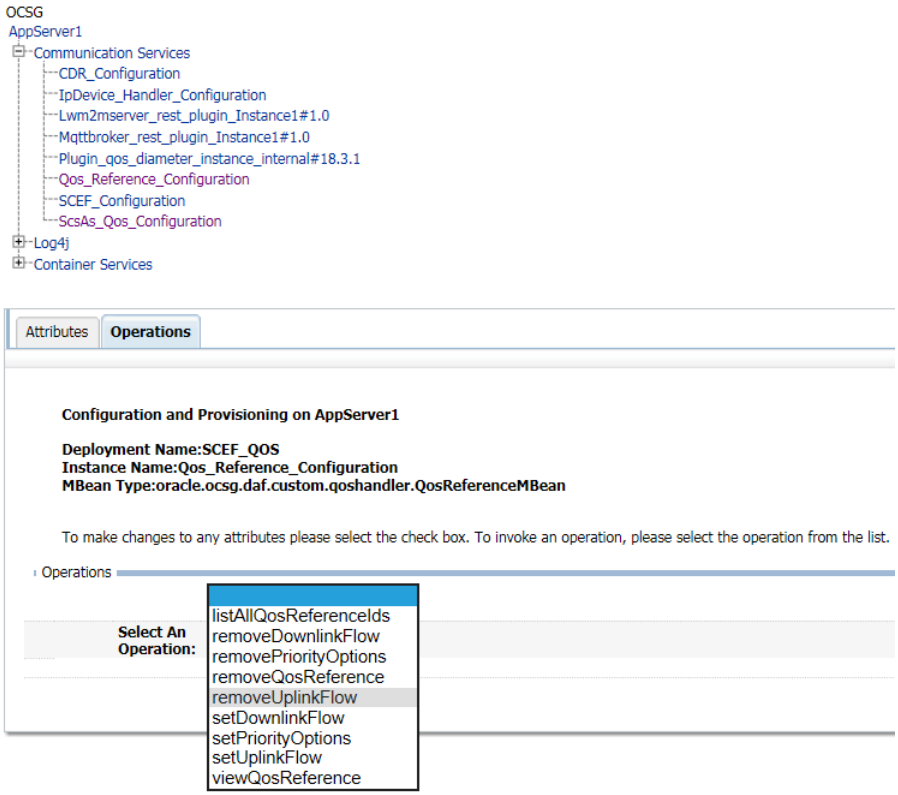
Manage QoS Reference Configuration

QoS reference configuration contains configurations related to up/downlink flow description and priority settings of session for this Qos Reference. A QoS reference configuration is uniquely identified by QoS Reference id. In order to select a QoS reference configured in the system the QoS Reference id can either be sent in T8 request or it can be configured for an SCS as default QoS reference to be used. Currently this configuration allows user to set(add/modify), remove, view the up/downlink data flows and priority options. Below are the steps to manage this configuration.

> **Note:**
>
> Any change made to an Application Server (AppServer) reflects to all other servers.

1. Access the DSR API GW Admin console using https://<Admin-Server-XMI-IP>:9002/console
2. Login using the admin account created when configuring the API GW. The default username is **weblogic**
3. Navigate to **OCSG > AppServerx > Communication Services > Qos_Reference_Configuration > Operations** (tab)

Figure A-26 Manage QoS Reference Configuration



Add/Modify Priority Option

Add/Modify the priority option:

1. Select **setPriorityOptions**.
2. Enter valid values for these fields:
   - Reservation Priority
   - MPS Identifier
   - Service URN
3. Click **Invoke**.

Remove Priority Option

Remove the priority option:

1. Select **removePriorityOptions**.
2. Enter the **Qos Reference ID** for which the priority options needs to be removed and click **Invoke**.

Add/Modify Uplink Flow

Add/Modify the uplink flow:

1. Select **setUplinkFlow**.

2. Enter valid values for these fields:
   - Qos Reference ID
   - Uplink Protocol
   - Uplink Source Address
   - Uplink Source Port
   - Uplink Destination Address
   - Uplink Destination Port
3. Click **Invoke**.

Remove Uplink Flow

Remove the uplink flow option:

1. Select **removeUplinkFlow**.
2. Enter the **Qos Reference ID** for which the uplink flow needs to be removed and click **Invoke**.

Add/Modify Downlink Flow

Add/Modify the downlink flow option:

1. Select **setDownlinkFlow**.
2. Enter valid values for these fields:
   - Qos Reference ID
   - Downlink Protocol
   - Downlink Source Address
   - Downlink Source Port
   - Downlink Destination Address
   - Downlink Destination Port
3. Click **Invoke**.

Remove Downlink Flow

Remove the downlink flow option:

1. Select **removeDownlinkFlow**.
2. Enter the **Qos Reference ID** for which the downlink flow needs to be removed and click **Invoke**.

View QoS Reference Flow

View the QoS reference flow:

1. Select **viewQosReference**.
2. Enter the **Qos Reference ID** for which the configured up/downlink flow and priority options need to be viewed and click **Invoke**.

List All QoS Reference IDs Configured

List all the QoS Reference IDs that are configured:

1. Select **listAllQosReferenceIds** operation and click **Invoke**.

Configure QoS Control API in DSR API Gateway

QoS Control APIs needs to be configured in the DSR API gateway before the partner application subscribes for it. Here are the steps to configure the APIs in DSR API Gateway:

1. Create API
2. Configure QoS Action to QoS Control API
3. Publish the API

Create API

Run this curl command to create the API:

```
curl -X POST -u <partner and API management portal username> -k https://<App
Server IP>:9002/portal/prm/prm_pm_rest/services/prm_pm/services/
partner_manager/api/PartnerManagerApi/createAPI -H 'Content-Type:
application/json' -H 'cache-control: no-cache' -d '{"createAPI":{"apiObject":
{"apiDisplayName":"3gpp-as-session-with-qos","apiName":"apiroot-
qos","apiVersion":"v1","apiInterfaces":[{"name":"apiroot-qos-
vv1","apiMethods":
[{"name":"as_session_with_qos_post","displayName":"as_session_with_qos_post",
"path":"3gpp-as-session-with-qos/v1/{scsAsId}/
subscriptions/","httpVerb":"POST","servicePath":"3gpp-as-session-with-qos/v1/
{scsAsId}/subscriptions/","serviceHttpVerb":"POST","expose":true},
{"name":"as_session_with_qos_put","displayName":"as_session_with_qos_put","pa
th":"3gpp-as-session-with-qos/v1/{scsAsId}/subscriptions/
{subscriptionId}","httpVerb":"PUT","servicePath":"3gpp-as-session-with-
qos/v1/{scsAsId}/subscriptions/
{subscriptionId}","serviceHttpVerb":"PUT","expose":true},
{"name":"as_session_with_qos_delete","displayName":"as_session_with_qos_delet
e","path":"3gpp-as-session-with-qos/v1/{scsAsId}/subscriptions/
{subscriptionId}","httpVerb":"DELETE","servicePath":"3gpp-as-session-with-
qos/v1/{scsAsId}/subscriptions/
{subscriptionId}","serviceHttpVerb":"DELETE","expose":true}],"displayName":"a
piroot-qos-vv1","fileLocation":"https://portal.3gpp.org/desktopmodules/
Specifications/SpecificationDetails.aspx?specificationId=3239"}],"wadlFiles":
[],"northBoundWadlFiles":[],"description":"Setup and manage AS session with
required QoS ","facade":"REST","direction":"AOMT","serviceType":"by-
url","protocol":"http://10.10.10.10","privilege":0,"link":"https://
portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?
specificationId=3239","accessType":"BOTH","apiAuthTypes":
["TEXT","OAUTH"],"authType":"NONE","authToken":"","networkProxy":"","networkA
uthorizationURI":"","networkTokenURI":"","networkClientRedirectURI":"","group
s":[],"icon":"expressive/api.png"}}}'
```

> **Note:**
>
> Content within <> must be filled with appropriate value.

Example response:

```
{"createAPIResponse":{"apiId":"77474356-2983-422a-9074-37d602786726"}}
```

Configure QoS Action to QoS Control API

Run this command to configure action:

```
curl -X POST -u <partner and API management portal username> -k
https://<App Server IP>:9002/portal/prm/prm_pm_rest/services/prm_pm/
services/partner_manager/actionchain/submitActionChain -H 'Content-
Type: application/json' -H 'cache-control: no-cache' -d
'{"submitActionChain":{"apiDisplayName":"3gpp-as-session-with-
qos","apiId":"<apiId received in createAPIResponse
response>","requestActions":[{"name":"SCEFCustomQos","content":"<?xml
version=\"1.0\" encoding=\"UTF-8\" standalone=\"yes\"?
><scefCustomQosActionConfig><instanceId>1234</instanceId></
scefCustomQosActionConfig>"}],"responseActions":[],"configVersion":1}}'
```

> **Note:**
>
> Content within <> must be filled with appropriate value.

Example response:

```
{"submitActionChainResponse":{}}
```

Publish the API

Run this command to PUBLISH the API:

```
curl -X POST -u <partner and API management portal username> -k
https://<App Server IP>:9002/portal/prm/prm_pm_rest/services/prm_pm/
services/partner_manager/api/PartnerManagerApi/updateApiStatus -H
'Content-Type: application/json' -H 'cache-control: no-cache' -d
'{"updateApiStatus":{"apiName":"<apiId received in createAPIResponse
response>","apiVersion":"v1","status":"PUBLISHED"}}'
```

> **Note:**
>
> Content within <> must be filled with appropriate value.

Example response:

```
{"updateApiStatusResponse":{}}
```

The newly created API displays on the Partner and API Management portal.

Modify Log4j2config.xml

Modify log4j2config.xml in all AppServers to change status from `trace` to `info`, that is,

```
<Configuration monitorInterval="5"
packages="oracle.ocsg.daf.custom.action.customlog4j"
status="trace">
```

to

```
<Configuration monitorInterval="5"
packages="oracle.ocsg.daf.custom.action.customlog4j" status="info">
```

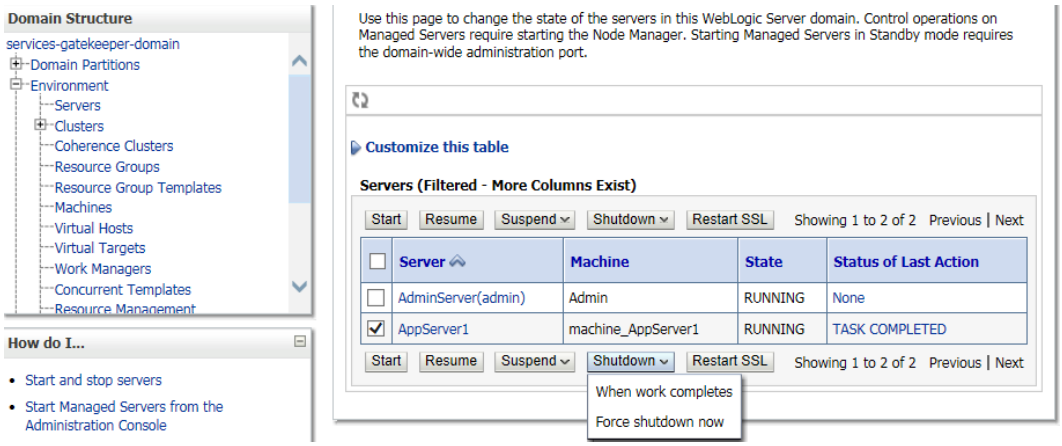
This xml file is available under the location: `/u03/app/oracle/ocsg-18.3.1/user_projects/domains/services-gatekeeper-domain/log4j`

After the above change is done on all AppServers, restart all AppServers from the Admin console:

```
https://<floating ip of admin server>:9002/console/login/LoginForm.jsp
```

Steps to Restart AppServers

1. Log into the Admin console.
2. From the side menu, navigate to **Domain Structure**, and then **Environment**, and then **Servers**.
3. From the Control tab, mark the checkboxes next to each AppServer and click **Shutdown**, and then **Force shutdown now**.



Click **Yes** to confirm.

4. Wait until the state of all AppServer changes to SHUTDOWN.

5. Mark the checkboxes next to each AppServer and click **Start**.
6. Click **Yes** to confirm.

Provisioning OCSG

This section describes how to provision the Oracle Communications Services Gatekeeper (OCSG).

Provisioning the OCSG involves these steps:

- Expose API URLs
- On Boarding a Partner
  - Register a Partner Account
  - Approve (or Reject) a Partner Account
  - Create a Partner Group
  - Assign a Partner to a Group
  - Create a Partner Application
  - Approve (or Reject) an Application Creation
  - Set Application Password

Expose API URLs

The DSR API GW exposes 3GPP T8 resource URLs over the XMI subnet and port 10002 for HTTPS traffic. The deployment topology mandates using an external load balance, which should be owned and maintained by the customer. The load balancer is configured to send HTTPS traffic to all DSR API GW AppServers belonging to the site over XMI IP and port 10002.

Each resource URL format of the T8 API specification is prefixed with /<apiroot-{nidd/me/dt/ecr}>/v1.

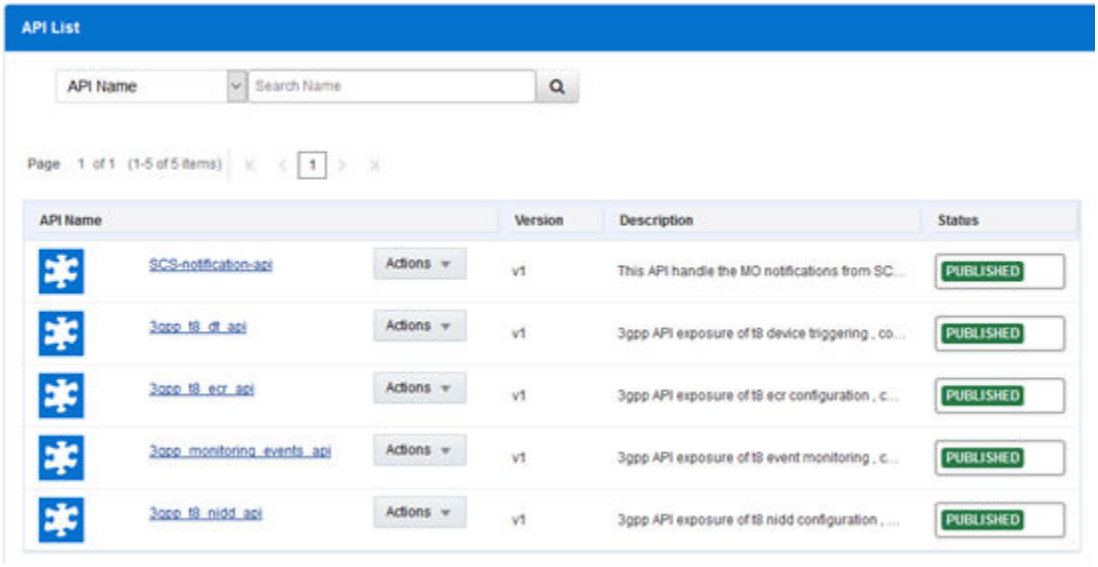
Apiroot is the property provided while configuring DSR API GW.

For example, the NIDD configuration URL will be where the apiroot provided is *operator1*:

```
/operator1-nidd/v1/3gpp_t8_nidd/v1/{scsAsId}/configurations
```

The T8 APIs for NIDD, Monitoring Events, Device Triggering, and ECR are defined on the Partner and API Management portal, which can be accessed using the operator account.

Figure A-27 Expose API URLs



On Boarding a Partner

This procedure on-boards an SCS/AS into SCEF-OCSG.

1. Register a partner account.
2. Approve partner account creation.
3. Create a partner group.
4. Assign the partner to a specific group.
5. Create a partner application.
6. Approve the application creation request.
7. Set the traffic password for the application.

The procedures that follow in this section explain each step in more detail.

Register a Partner Account

A partner account can be registered using a self registration process or you can register an account for the partner. Both processes can also be done using a GUI interface or the REST interface. All procedures are described in this section. Refer to Diameter Signaling Router SCEF Partner User's Guide for configuration information using the GUI.

Self Registration Using REST

To self register as a partner, use the POST method from the /prm_pm_rest/services/prm_pr/services/register/Register/registers resource URL.

For more details on the request and response formats, see https://docs.oracle.com/communications/E81149_01/doc.70/e96582/resource_Partner_Account.html#resource_Partner_Account_registerSP_POST.

An example of a self registration request and response follow:

Request:

```
POST /prm_pm_rest/services/prm_pr/services/register/Register/
registerSP HTTP/1.1
Host: 10.178.254.224:9001
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/
20100101 Firefox/52.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json
x-requested-with: XMLHttpRequest
Referer: http://10.178.254.224:9001/portal/partner/index/register.html
Content-Length: 650
Connection: keep-alive

{"registerSP":{"spInfo":
{"userName":"test_user1","emailAddr":"test_user1@oracle.com",
"password":"password123”,"phone":"91984538533","securityAnswerChoice":0
,
"securityAnswer":"tp1","firstName":"test_fn","lastName":"test_ln","comp
any":"oracle",
"companyURL":"http://
oracle.com","stateOrProvince":"Karnataka","zipOrPostalCode":"560072",
"streetAddress":"kudebeesanhalli","city":"Bangalaore","country":"India"
,
"contacts":
[{"city":"Bangalore","contactTimeFrom":"09:00","contactTimeTo":"17:00",
"country":"India","emailAddress":"test_partner@oracle.com","firstName":
"test_fn",
"lastName":"test_ln"}],"userType":"PRM_SP"}}}
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 30 Aug 2018 08:40:13 GMT
Content-Length: 25
Content-Type: application/json

{"registerSPResponse":{}}
```

Register an Account for a Partner Using the GUI

1. Access the partner and API management portal at https://<AppServerx-XMI-IP>:9002/portal/partner/index/partnerLogin.html.
2. Login with the operator account.
3. Click on the Partners tab.
4. Click **Create Partner Account**.
5. Provide required details.
6. Click **Create Partner**.

Register an Account for a Partner Using REST

To register a partner, use the POST method from the /portal/prm/prm_pm_rest/services/accountmanage/AccountManagement/createUser resource URL.

To authorize the request, use the operator username and password in the header of the request.

For more details on the request and response formats, see https://docs.oracle.com/communications/E81149_01/doc.70/e96582/resource_Partner_Manager_Account.html.

An example of a registration request and response follow:

Request:

```
POST /portal/prm/prm_pm_rest/services/accountmanage/AccountManagement/
createUser HTTP/1.1
Host: 10.75.244.188:9001
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:57.0) Gecko/20100101
Firefox/57.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.75.244.188:9001/portal/partner-manager/index/main.html
Content-Type: application/json
AuthorizationX: Basic
b3A6e0FFU310bnp2bmliSEpPbWNBdlo5QXZ0d3pzSFoxZVBBZnhNWEY3eG9CZ1kzci9ZPQ==
X-Requested-With: XMLHttpRequest
Content-Length: 671
Cookie:
ADMINCONSOLESESSION=645Fdnqjm2LmBV4xf2yqeM9Tk3YngS0_q7GwCrZMWfNPiXsLe9xI!-676
027630
Connection: keep-alive

{"createUser":{"userInfo":
{"userName":"testuser","emailAddr":"user1@usercompany.com",
"password":"password1234","phone":"9845398765","secureityAnswerChoice":0,
"secureityAnswer":"user1","firstName":"userfirstname","lastName":"userlastnam
e",
"company":"usercompany","companyURL":"http://
usercompany.com","stateOrProvince":"state1",
"zipOrPostalCode":"560037","streetAddress":"street1","city":"bangalore","coun
try":"India",
"contacts":
[{"city":"bangalore","contactTimeFrom":"08:00","contactTimeTo":"17:00",
"country":"India","emailAddress":"user1@usercompany.com","firstName":"userfir
stname",
"lastName":"userlastname"}],"userType":"PRM_SP"}}}
```

Response:

```
HTTP/1.1 200 OK
Content-Length: 0Server: Jetty(8.0.1.0)
```

Approve (or Reject) a Partner Account

When a partner account is created using the self-registration process on the partner portal, the operator needs to approve (or reject) the request. This can be done using either the GUI interface or REST interface.

Approve a Partner Account Using the GUI

1. Access the partner and API management portal at https://<AppServerx-XMI-IP>:9002/portal/partner/index/partnerLogin.html.
2. Login with the operator account.
3. Click on the red circle on top right corner.
   The screen displays all requests pending an approval.
4. Select the partner request and right click to View Details.
5. Review the details and approve or reject the request.

Approve a Partner Account Using REST

The partner approval using REST is a three step process involving getting the notification, approving the request, and updating the notification status.

1. To get the notification, use the GET method from the /portal/prm/prm_pm_rest/services/partner_manager/notification/PartnerManagerNotification/listNotificationsByStatus/UNREAD resource URL.
   To authorize the request, use the operator username and password in the header of the request.

   For more details on the request and response formats, see https://docs.oracle.com/communications/E81149_01/doc.70/e96582/resource_Partner_Manager_Notification.html.

   An example of the notification request and response follow:

   Request:

```
GET /portal/prm/prm_pm_rest/services/partner_manager/notification/
PartnerManagerNotification/listNotificationsByStatus/UNREAD HTTP/1.1
Host: 10.75.244.188:9001
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:58.0) Gecko/
20100101 Firefox/58.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.75.244.188:9001/portal/partner/index/
partnerMain.html
AuthorizationX: Basic
cGFydG5lcjE6e0FFU301Z2Q1UFpwcDZVTmJHdkxrRnBPSXpuMDNMNGxNeDRDZUpCcnBQ
TjJYaXZrPQ==
X-Requested-With: XMLHttpRequest
Connection: keep-alive
```

Response:

```
HTTP/1.1 200 OK

Date: Wed, 31 Jan 2018 05:50:57 GMT
Content-Length: 9447
Content-Type: application/json
X-FRAME-OPTIONS: DENY

{"ListNotificationsByStatusResponse":{"return":
[{"id":"113e1904-8419-4cd2-958a-659a950aba6c",
"content":"Partner Create Application
Task","date":"01/31/2018","receiver":"PM",
"sender":"partner1","senderCompany":"oracle","redirectObject":
{"type":"ns4:application",
"notificationId":"113e1904-8419-4cd2-958a-659a950aba6c","applicationID":"f
a789e8d-9b97-455a-b556-491ed5253da5",
"applicationName":"mmitestapp1","partnerName":"partner1","partnerCompany":
"oracle",
"description":"mmi testing application 1","applicationAPIs":
[{"apiDisplayName":"3gpp_t8_nidd",
"apiName":"3gpp_t8_nidd","accessURL":"http://10.10.10.9:8001/3gpp_t8_nidd/
v1\nhttps://10.10.10.9:7002/3gpp_t8_nidd/v1",
"apiVersion":"v1","apiDescription":"dsr nidd test api
","needReadContract":false}],
"trafficUser":"partner1_mmitestapp1","trafficPassword":
"{AES}UEYOH2WTIo5Kwgodl7uFtmMCr5oLLBTz3H6jQ4jK5j9AoPdBhjYJiqtqpvB86IuYwNyb
nTP+x3hTAgn/UTLrUw==",
"submitDate":"2018-01-31-05:00","effectiveFrom":"2018-01-31-05:00","effect
iveTo":"2018-03-28-04:00",
"status":"CREATE PENDING APPROVAL","lockStatus":"UNLOCKED","quota":
{"days":1,"limitExceedOK":true,
"qtaLimit":10000},"rate":
{"reqLimit":10,"timePeriod":1},"icon":"expressive/app.png"},
"status":"UNREAD"},{"id":"9bdc365a-40fd-496c-9cff-
dd1ec805dd18","content":"Partner Delete Pending Application Task",
"date":"01/31/2018","receiver":"PM","sender":"partner1","senderCompany":"o
racle",
"redirectObject":
{"type":"ns4:application","notificationId":"9bdc365a-40fd-496c-9cff-
dd1ec805dd18",
"applicationID":"60c5e194-ad46-4935-
ba78-4777bab65eaf","applicationName":"mmitestapp1",
"partnerName":"partner1","partnerCompany":"oracle","description":"mmi
testing application 1",
"applicationAPIs":
[{"apiDisplayName":"3gpp_t8_nidd","apiName":"3gpp_t8_nidd",
"accessURL":"http://10.10.10.9:8001/3gpp_t8_nidd/v1\nhttps://
10.10.10.9:7002/3gpp_t8_nidd/v1",
"apiVersion":"v1","apiDescription":"dsr nidd test api
","applicationMethodSLAs":[{"methodName":"",
"interfaceName":"57cf5ce0-a175-43d2-a1f4-53fb3ebae851",
"quota":{"days":0,"limitExceedOK":false,"qtaLimit":0},"rate":
{"reqLimit":0,"timePeriod":0},
"methodGuarantee":
```

```
{"reqLimitGuarantee":0,"timePeriodGuarantee":0}}],"needReadContract"
:false}],
"trafficUser":"partner1_mmitestapp1","submitDate":"2018-01-31-05:00"
,"effectiveFrom":"2018-01-31-05:00",
"effectiveTo":"2018-04-24-04:00","status":"CREATE PENDING
APPROVAL","lockStatus":"UNLOCKED",
"quota":{"days":1,"limitExceedOK":true,"qtaLimit":10000},"rate":
{"reqLimit":10,"timePeriod":1},
"icon":"expressive/app.png"},"status":"UNREAD"},{"id":"b012db05-
caac-421f-ad3b-95d5350fc72a",
"content":"Partner Create Application
Task","date":"01/31/2018","receiver":"PM","sender":"partner1",
"senderCompany":"oracle","redirectObject":{"type":"ns4:application",
"notificationId":"b012db05-caac-421f-
ad3b-95d5350fc72a","applicationID":"60c5e194-ad46-4935-
ba78-4777bab65eaf",
"applicationName":"mmitestapp1","partnerName":"partner1","partnerCom
pany":"oracle",
"description":"mmi testing application 1","applicationAPIs":
[{"apiDisplayName":"3gpp_t8_nidd",
"apiName":"3gpp_t8_nidd","accessURL":"http://
10.10.10.9:8001/3gpp_t8_nidd/v1\nhttps://
10.10.10.9:7002/3gpp_t8_nidd/v1",
"apiVersion":"v1","apiDescription":"dsr nidd test api
","needReadContract":false}],
"trafficUser":"partner1_mmitestapp1","trafficPassword":
"{AES}pqXqICn4W4IJq/u8kitcbn8w82RJKQKZbI2WUaV9KzKOMOcxSUQhU1vd/
9hEsZcDBwqjP93HllvhoU41UwOCaw==",
"submitDate":"2018-01-31-05:00","effectiveFrom":"2018-01-31-05:00","
effectiveTo":"2018-04-24-04:00",
"status":"CREATE PENDING APPROVAL","lockStatus":"UNLOCKED","quota":
{"days":1,"limitExceedOK":true,"qtaLimit":10000},
"rate":{"reqLimit":10,"timePeriod":1},"icon":"expressive/
app.png"},"status":"UNREAD"}]}}
```


2. To approve the partner account creation request, use the POST method from the /portal/prm/prm_pm_rest/services/accountmanage/AccountManagement/approve resource URL.
   Note the account creation notification ID from the previous step.

   To authorize the request, use the operator username and password in the header of the request.

   For more details on the request and response formats, see https://docs.oracle.com/communications/E81149_01/doc.70/e96582/resource_Partner_Manager_Account.html.

   An example of an approval request and response follow:

   Request:


```
POST /portal/prm/prm_pm_rest/services/accountmanage/
AccountManagement/approve HTTP/1.1
Host: 10.178.254.224:9001
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/
20100101 Firefox/52.0
Accept: application/json, text/javascript, */*; q=0.01
```

```
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json
AuthorizationX: Basic
b3JhY2xlb3AxOntBRVN9ZnBTRHBaeWw0dGRqR0lob3c2SzZFOThGT2tKbGxyZXB5Y3RrbUx0Mm
hPWT0=
x-requested-with: XMLHttpRequest
Referer: http://10.178.254.224:9001/portal/partner-manager/index/main.html
Content-Length: 935
Connection: keep-alive

{"approve":{"userInfo":
{"city":"Bangalaore","company":"oracle","companyURL":"http://oracle.com",
"contacts":
[{"city":"Bangalore","contactTimeFrom":"09:00","contactTimeTo":"17:00","co
untry":"India",
"emailAddress":"test_partner@oracle.com","firstName":"test_fn","lastName":
"test_ln","phone":null}],
"country":"India","emailAddr":"test_user1@oracle.com","financial":
{"bankAccountNumber":"",
"bankAddress":"","bankName":"","bankRoutingNumber":"","city":"","country":
"","invoiceTo":"",
"referenceAccount":"","stateOrProvince":"","taxID":"","zipOrPostalCode":""
},"firstName":"test_fn",
"lastName":"test_ln","password":"{AES}mhY96ryJA82JHEiChWJo3rDczngO/
YuMYN5tSxH4Oko=","phone":"91984538533",
"securityAnswer":"tp1","securityAnswerChoice":"0","stateOrProvince":"Karna
taka","status":0,
"streetAddress":"kudebeesanhalli","userName":"test_user1","zipOrPostalCode
":"560072",
"userType":"PRM_SP","notificationId":"b1e706fb-4436-401f-972c-99821f052805
"}}}
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 30 Aug 2018 08:42:48 GMT
Content-Length: 22
Content-Type: application/json
X-Frame-Options: DENY

{"approveResponse":{}}
```

3. Once the request has been approved or rejected, change the notification status, using the POST method from the /portal/prm/prm_pm_rest/services/partner_manager/notification/PartnerManagerNotification/updateNotificationStatus resource URL.
To authorize the request, use the operator username and password in the header of the request.

   For more details on the request and response formats, see https://docs.oracle.com/communications/E81149_01/doc.70/e96582/resource_Partner_Manager_Notification.html.

   An example of an updated notification status request and response follow:

Request:

```
POST /portal/prm/prm_pm_rest/services/partner_manager/notification/
PartnerManagerNotification/updateNotificationStatus HTTP/1.1
Host: 10.178.254.224:9001
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/
20100101 Firefox/52.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json
AuthorizationX: Basic
b3JhY2xlb3AxOntBRVN9ZnBTRHBaeWw0dGRqR0lob3c2SzZFOThGT2tKbGxyZXB5Y3Rr
bUx0MmhPWT0=
x-requested-with: XMLHttpRequest
Referer: http://10.178.254.224:9001/portal/partner-manager/index/
main.html
Content-Length: 102
Connection: keep-alive

{"updateNotificationStatus":
{"notificationId":"b1e706fb-4436-401f-972c-99821f052805","status":"R
EAD"}}
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 30 Aug 2018 08:42:48 GMT
Content-Length: 39
Content-Type: application/json
X-Frame-Options: DENY

{"updateNotificationStatusResponse":{}}
```

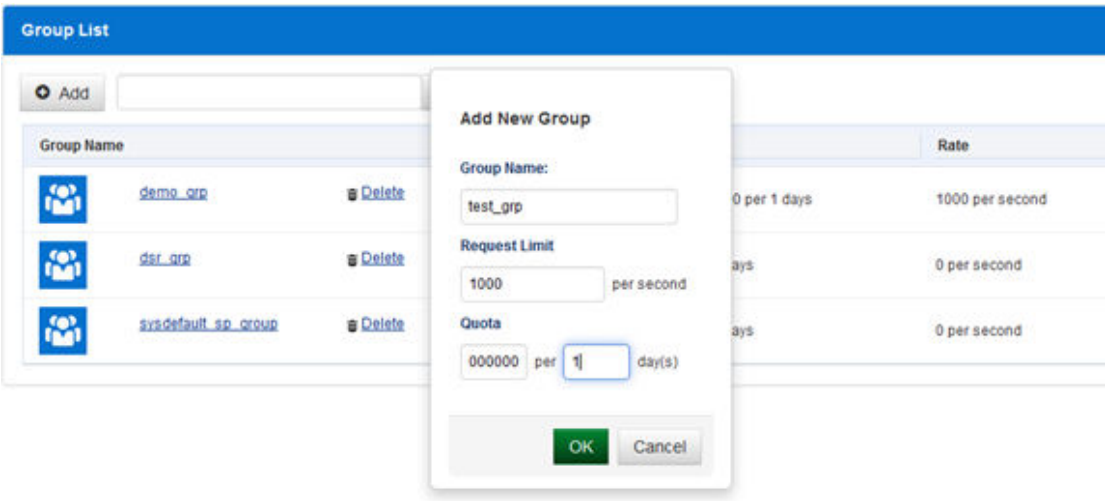
Create a Partner Group

Once a partner account is created and approved, it is added to a group. To create a group, use either the GUI interface or the REST interface. Both procedures are described in this section.

Create a Partner Group Using the GUI

1. Access the partner and API management portal at https://<AppServerx-XMI-IP>:9002/portal/partner/index/partnerLogin.html.
2. Login with the operator account.
3. Click on the Partners tab.
4. Click **Groups**.
5. Click **Add**.
6. Type the *Group Name*, *Request Limit*, and *Quota* allowed for the partner group.
7. Click **OK**.

Figure A-28 Create Partner Group



Create a Partner Group Using REST

To create a partner group, use the POST method from the /portal/prm/prm_pm_rest/services/partner_manager/group/PartnerManagerSlaGroup/createServiceProviderGroup resource URL.

To authorize the request, use the operator username and password in the header of the request.

For more details on the request and response formats, see https://docs.oracle.com/communications/E81149_01/doc.70/e96582/resource_Group.html.

An example to create a partner group request and response follow:

Request:

```
POST /portal/prm/prm_pm_rest/services/partner_manager/group/
PartnerManagerSlaGroup/createServiceProviderGroup HTTP/1.1
Host: 10.75.244.188:9001
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:57.0) Gecko/20100101
Firefox/57.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.75.244.188:9001/portal/partner-manager/index/main.html
Content-Type: application/json
AuthorizationX: Basic
b3A6e0FFU31UT0RyWXN5dE0yMWJMZ1VndVJTVTJWbk1XV3FSaFFNRlBLRDhaVG1RbHdJPQ==
X-Requested-With: XMLHttpRequest
Content-Length: 143
Cookie:
ADMINCONSOLESESSION=645Fdnqjm2LmBV4xf2yqeM9Tk3YngS0_q7GwCrZMWfNPiXsLe9xI!-676
027630
Connection: keep-alive

{"createServiceProviderGroup":{"groupName":"user2_group","rate":
```

```
{"reqLimit":"10000","timePeriod":1},"quota":
{"qtaLimit":"1000000","days":"1"}}}
```

Response:

```
HTTP/1.1 200 OK
Date: Tue, 30 Jan 2018 06:46:09 GMT
Content-Length: 41
Content-Type: application/json
X-FRAME-OPTIONS: DENY

{"createServiceProviderGroupResponse":{}}
```

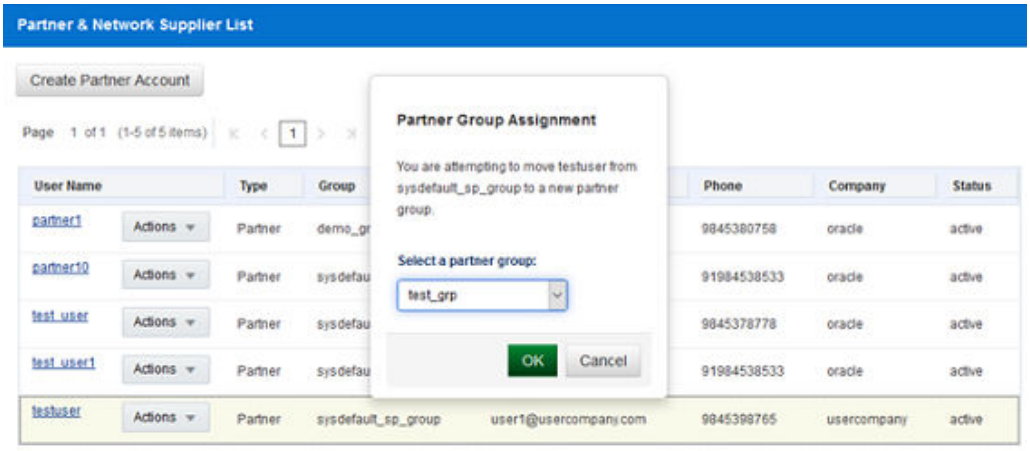
Assign a Partner to a Group

After a partner account and group have been created, the operator needs to add the account to the group. This can be done using either the GUI interface or REST interface.

Assign a Partner to a Group Using the GUI

1. Access the partner and API management portal at https://<AppServerx-XMI-IP>:9002/portal/partner/index/partnerLogin.html.
2. Login with the operator account.
3. Click on the Partners tab.
4. Select *Assign Group* from the Actions options.
5. Select the group to which the partner is to be assigned.
6. Click **OK**.

Figure A-29 Add Partner to Group



Assign a Partner to a Group Using REST

To assign a partner to a group, use the POST method from the /portal/prm/prm_pm_rest/services/partner_manager/group/PartnerManagerSlaGroup/confirmMovePartnerToGroup resource URL.

To authorize the request, use the operator username and password in the header of the request.

For more details on the request and response formats, see https://docs.oracle.com/communications/E81149_01/doc.70/e96582/resource_Group.html.

An example of how to add a partner to a group request and response follow:

Request:

```
POST /portal/prm/prm_pm_rest/services/partner_manager/group/
PartnerManagerSlaGroup/confirmMovePartnerToGroup HTTP/1.1
Host: 10.75.244.188:9001
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:57.0) Gecko/20100101
Firefox/57.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.75.244.188:9001/portal/partner-manager/index/main.html
Content-Type: application/json
AuthorizationX: Basic
b3A6e0FFU31UT0RyWXN5dE0yMWJMZ1VndVJTVTJWbk1XV3FSaFFNRlBLRDhaVG1RbHdJPQ==
X-Requested-With: XMLHttpRequest
Content-Length: 107
Cookie:
ADMINCONSOLESESSION=645Fdnqjm2LmBV4xf2yqeM9Tk3YngS0_q7GwCrZMWfNPiXsLe9xI!-676
027630
Connection: keep-alive

{"confirmMovePartnerToGroup":
{"partnerName":"testuser","newGroupName":"user1_group","action":"EXPAND_SLA"}
}
```

Response:

```
HTTP/1.1 200 OK
Date: Tue, 30 Jan 2018 06:50:24 GMT
Content-Length: 40
Content-Type: application/json
X-FRAME-OPTIONS: DENY

{"confirmMovePartnerToGroupResponse":{}}
```

Create a Partner Application

To access APIs exposed by the DSR API GW, a logical unit partner application is created by the partner. This can be done using either the GUI interface or REST interface. Refer to

Diameter Signaling Router SCEF Partner User's Guide for configuration information using the GUI.

Create a Partner Application Using REST

To create a partner application, use the POST method from the /portal/prm/prm_pm_rest/services/prm_pm/services/partner/application/PartnerApplication/createApplication resource URL.

To authorize the request, use the partner username and password in the header of the request.

For more details on the request and response formats, see https://docs.oracle.com/communications/E81149_01/doc.70/e96582/resource_Partner_Application.html.

An example of how to create a partner application request and response follow:

Request:

```
POST /portal/prm/prm_pm_rest/services/prm_pm/services/partner/
application/PartnerApplication/createApplication HTTP/1.1
Host: 10.178.254.224:9001
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/
20100101 Firefox/52.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json
AuthorizationX: Basic
cGFydG5lcjE6e0FFU31GYjZmNWt2UmRRZEVyU2FzR08zL0kwYlc0aG1sdVE3SDZ4YktIN0p
kZ2w4PQ==
x-requested-with: XMLHttpRequest
Referer: http://10.178.254.224:9001/portal/partner/index/
partnerMain.html
Content-Length: 545
Connection: keep-alive

{"createApplication":{"application":
{"applicationName":"test_app_3","description":"testing application 3",
"trafficUser":"partner1_test_app1_user","trafficPassword":"password1234
","effectiveFrom":"2018-08-31",
"effectiveTo":"2019-04-19","partnerName":"partner1","quota":
{"days":"1","limitExceedOK":true,"qtaLimit":"100000"},
"rate":{"reqLimit":"10","timePeriod":"1"},"applicationAPIs":
[{"apiName":"80cec996-02a1-40ec-bb66-fece0b86b317"},
{"apiName":"36b271ee-0d37-46b7-96c2-72deec416bee"},
{"apiName":"9d5c8500-d5cc-4fda-8d8c-83bec26141a7"},
{"apiName":"19a44504-fb39-477f-b920-49f5bf85f443"}],"icon":"expressive/
app.png"}}}
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 30 Aug 2018 09:56:55 GMT
Content-Length: 86
Content-Type: application/json
```

```
X-Frame-Options: DENY

{"createApplicationResponse":{"applicationID":"ec8fa535-f45c-42fd-a4b6-
bcf1a584be82"}}
```

API Names are IDs generated by DSR API GW dynamically. A partner can retrieve the API IDs by using REST from the /portal/prm/prm_pm_rest/services/prm_pm/services/partner/api/PartnerApi/getAPIs resource URL.

Refer to https://docs.oracle.com/communications/E81149_01/doc.70/e96582/resource_Partner_API.html for more information.

Approve (or Reject) an Application Creation

When a partner application is created, the operator needs to approve (or reject) the request. This can be done using either the GUI interface or REST interface.

Approve a Partner Application Using the GUI

1. Access the partner and API management portal at https://<AppServerx-XMI-IP>:9002/portal/partner/index/partnerLogin.html.
2. Login with the operator account.
3. Click on the red circle on top right corner.
   The screen displays all requests pending an approval.
4. Select the application creation request and right click to View Details.
5. Review the details and approve or reject the application request.

Approve a Partner Application Using

The partner application approval using REST is a three step process involving getting the notification, approving the request, and updating the notification status.

1. To get the notification, use the GET method from the /portal/prm/prm_pm_rest/services/partner_manager/notification/PartnerManagerNotification/listNotificationsByStatus/UNREAD resource URL.
   To authorize the request, use the operator username and password in the header of the request.

   For more details on the request and response formats, see https://docs.oracle.com/communications/E81149_01/doc.70/e96582/resource_Partner_Manager_Notification.html.

   An example of the notification request and response follow:

   Request:

   ```
   GET /portal/prm/prm_pm_rest/services/partner_manager/notification/
   PartnerManagerNotification/listNotificationsByStatus/UNREAD HTTP/1.1
   Host: 10.75.244.188:9001
   User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:58.0) Gecko/
   20100101 Firefox/58.0
   Accept: application/json, text/javascript, */*; q=0.01
   Accept-Language: en-US,en;q=0.5
   Accept-Encoding: gzip, deflate
   Referer: http://10.75.244.188:9001/portal/partner/index/partnerMain.html
   ```

```
AuthorizationX: Basic
cGFydG5lcjE6e0FFU301Z2Q1UFpwcDZVTmJHdkxrRnBPSXpuMDNMNGxNeDRDZUpCcnBQ
TjJYaXZrPQ==
X-Requested-With: XMLHttpRequest
Connection: keep-alive
```

Response:

```
HTTP/1.1 200 OK

Date: Wed, 31 Jan 2018 05:50:57 GMT
Content-Length: 9447
Content-Type: application/json
X-FRAME-OPTIONS: DENY

{"ListNotificationsByStatusResponse":{"return":
[{"id":"113e1904-8419-4cd2-958a-659a950aba6c",
"content":"Partner Create Application
Task","date":"01/31/2018","receiver":"PM",
"sender":"partner1","senderCompany":"oracle","redirectObject":
{"type":"ns4:application",
"notificationId":"113e1904-8419-4cd2-958a-659a950aba6c","application
ID":"fa789e8d-9b97-455a-b556-491ed5253da5",
"applicationName":"mmitestapp1","partnerName":"partner1","partnerCom
pany":"oracle",
"description":"mmi testing application 1","applicationAPIs":
[{"apiDisplayName":"3gpp_t8_nidd",
"apiName":"3gpp_t8_nidd","accessURL":"http://
10.10.10.9:8001/3gpp_t8_nidd/v1\nhttps://
10.10.10.9:7002/3gpp_t8_nidd/v1",
"apiVersion":"v1","apiDescription":"dsr nidd test api
","needReadContract":false}],
"trafficUser":"partner1_mmitestapp1","trafficPassword":
"{AES}UEYOH2WTIo5Kwgodl7uFtmMCr5oLLBTz3H6jQ4jK5j9AoPdBhjYJiqtqpvB86I
uYwNybnTP+x3hTAgn/UTLrUw==",
"submitDate":"2018-01-31-05:00","effectiveFrom":"2018-01-31-05:00","
effectiveTo":"2018-03-28-04:00",
"status":"CREATE PENDING APPROVAL","lockStatus":"UNLOCKED","quota":
{"days":1,"limitExceedOK":true,
"qtaLimit":10000},"rate":
{"reqLimit":10,"timePeriod":1},"icon":"expressive/app.png"},
"status":"UNREAD"},{"id":"9bdc365a-40fd-496c-9cff-
dd1ec805dd18","content":"Partner Delete Pending Application Task",
"date":"01/31/2018","receiver":"PM","sender":"partner1","senderCompa
ny":"oracle",
"redirectObject":
{"type":"ns4:application","notificationId":"9bdc365a-40fd-496c-9cff-
dd1ec805dd18",
"applicationID":"60c5e194-ad46-4935-
ba78-4777bab65eaf","applicationName":"mmitestapp1",
"partnerName":"partner1","partnerCompany":"oracle","description":"mm
i testing application 1",
"applicationAPIs":
[{"apiDisplayName":"3gpp_t8_nidd","apiName":"3gpp_t8_nidd",
```

```
"accessURL":"http://10.10.10.9:8001/3gpp_t8_nidd/v1\nhttps://
10.10.10.9:7002/3gpp_t8_nidd/v1",
"apiVersion":"v1","apiDescription":"dsr nidd test api
","applicationMethodSLAs":[{"methodName":"",
"interfaceName":"57cf5ce0-a175-43d2-a1f4-53fb3ebae851",
"quota":{"days":0,"limitExceedOK":false,"qtaLimit":0},"rate":
{"reqLimit":0,"timePeriod":0},
"methodGuarantee":
{"reqLimitGuarantee":0,"timePeriodGuarantee":0}}],"needReadContract":false
}],
"trafficUser":"partner1_mmitestapp1","submitDate":"2018-01-31-05:00","effe
ctiveFrom":"2018-01-31-05:00",
"effectiveTo":"2018-04-24-04:00","status":"CREATE PENDING
APPROVAL","lockStatus":"UNLOCKED",
"quota":{"days":1,"limitExceedOK":true,"qtaLimit":10000},"rate":
{"reqLimit":10,"timePeriod":1},
"icon":"expressive/app.png"},"status":"UNREAD"},{"id":"b012db05-caac-421f-
ad3b-95d5350fc72a",
"content":"Partner Create Application
Task","date":"01/31/2018","receiver":"PM","sender":"partner1",
"senderCompany":"oracle","redirectObject":{"type":"ns4:application",
"notificationId":"b012db05-caac-421f-
ad3b-95d5350fc72a","applicationID":"60c5e194-ad46-4935-ba78-4777bab65eaf",
"applicationName":"mmitestapp1","partnerName":"partner1","partnerCompany":
"oracle",
"description":"mmi testing application 1","applicationAPIs":
[{"apiDisplayName":"3gpp_t8_nidd",
"apiName":"3gpp_t8_nidd","accessURL":"http://10.10.10.9:8001/3gpp_t8_nidd/
v1\nhttps://10.10.10.9:7002/3gpp_t8_nidd/v1",
"apiVersion":"v1","apiDescription":"dsr nidd test api
","needReadContract":false}],
"trafficUser":"partner1_mmitestapp1","trafficPassword":
"{AES}pqXqICn4W4IJq/u8kitcbn8w82RJKQKZbI2WUaV9KzKOMOcxSUQhU1vd/
9hEsZcDBwqjP93HllvhoU41UwOCaw==",
"submitDate":"2018-01-31-05:00","effectiveFrom":"2018-01-31-05:00","effect
iveTo":"2018-04-24-04:00",
"status":"CREATE PENDING APPROVAL","lockStatus":"UNLOCKED","quota":
{"days":1,"limitExceedOK":true,"qtaLimit":10000},
"rate":{"reqLimit":10,"timePeriod":1},"icon":"expressive/
app.png"},"status":"UNREAD"}]}}
```


2. To approve the partner application creation request, use the POST method from the /portal/prm/prm_pm_rest/services/partner_manager/application/PartnerManagerApplication/updateCurrentSlaForApprove resource URL.
Note the application creation notification ID from the previous step.

   To authorize the request, use the operator username and password in the header of the request.

   For more details on the request and response formats, see https://docs.oracle.com/communications/E81149_01/doc.70/e96582/resource_Partner_Manager_Application.html.

   An example of an approval request and response follow:

Request:

```
POST /portal/prm/prm_pm_rest/services/partner_manager/application/
PartnerManagerApplication/updateCurrentSlaForApprove HTTP/1.1
Host: 10.75.244.188:9001
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:58.0) Gecko/
20100101 Firefox/58.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.75.244.188:9001/portal/partner-manager/index/
main.html
Content-Type: application/json
AuthorizationX: Basic
b3A6e0FFU303cnFtSmc2MGNSMW95S3NvSkZ1V01VOTZ5MlI1cXp6alhCdHNTRkVZTGRv
PQ==
X-Requested-With: XMLHttpRequest
Content-Length: 611
Connection: keep-alive

{"updateCurrentSlaForApprove":{"application":
{"notificationId":"92b78c6f-2f01-4ed6-8d01-75c43118d53e",
"applicationID":"b07e1e8e-0f85-4bdc-9422-1c4b88bbb51b","applicationN
ame":"mmi2","partnerName":"partner1",
"partnerCompany":"oracle","description":"mmi
2","trafficUser":"partner1_mmi2",
"trafficPassword":"{AES}XE1iY7gDD9sq3o7Ug1WAi+dgAXmnYu7LsYsCUTQYvbQ=
","submitDate":"2018-01-31-05:00",
"effectiveFrom":"2018-01-31-05:00","effectiveTo":"2018-04-10-04:00",
"status":"CREATE PENDING APPROVAL",
"lockStatus":"UNLOCKED","quota":
{"qtaLimit":1000,"limitExceedOK":false,"days":1},"rate":
{"reqLimit":10,"timePeriod":1}}}}
```

Response:

```
HTTP/1.1 200 OK
Date: Wed, 31 Jan 2018 05:58:20 GMT
Content-Length: 41
Content-Type: application/json
X-FRAME-OPTIONS: DENY

{"updateCurrentSlaForApproveResponse":{}}
```

3. Once the application request has been approved or rejected, change the notification status using the POST method from the /portal/prm/prm_pm_rest/services/partner_manager/notification/PartnerManagerNotification/updateNotificationStatus resource URL.
   To authorize the request, use the operator username and password in the header of the request.

   For more details on the request and response formats, see https://docs.oracle.com/communications/E81149_01/doc.70/e96582/resource_Partner_Manager_Notification.html.

An example of an updated notification status request and response follow:

Request:

```
POST /portal/prm/prm_pm_rest/services/partner_manager/notification/
PartnerManagerNotification/updateNotificationStatus HTTP/1.1
Host: 10.178.254.224:9001
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/20100101
Firefox/52.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json
AuthorizationX: Basic
b3JhY2xlb3AxOntBRVN9ZnBTRHBaeWw0dGRqR0lob3c2SzZFOThGT2tKbGxyZXB5Y3RrbUx0Mm
hPWT0=
x-requested-with: XMLHttpRequest
Referer: http://10.178.254.224:9001/portal/partner-manager/index/main.html
Content-Length: 102
Connection: keep-alive

{"updateNotificationStatus":
{"notificationId":"b1e706fb-4436-401f-972c-99821f052805","status":"READ"}}
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 30 Aug 2018 08:42:48 GMT
Content-Length: 39
Content-Type: application/json
X-Frame-Options: DENY

{"updateNotificationStatusResponse":{}}
```

Set Application Password

Password for partner application needs to be set by the partner, which is sent in the T8 API request to the DSR API GW. This process is needed only if creating the application using the GUI interface. If the REST interface was used to create the application, the password was set during the creation of the application.

1. Access the partner and API management portal at https://<AppServerx-XMI-IP>:9002/portal/partner/index/partnerLogin.html.
2. Log into the portal using the partner account.
3. Click on the Applications tab.
4. Select the application.
5. Click on the key symbol next to the Traffic User property.
6. Set the traffic password and click **Update**.

B
Error Codes

The following table lists the error codes for Device Status Query Troubleshooting APIs.

Table B-1 Error Codes

API Name	Error Code	Description
NIDD Troubleshooting API	404 - Not Found	When the device data is not found.
	500 - Internal Server Error	
	503 - Service Unavailable	DSR Connection Failure.
MONTE Non IP Troubleshooting API	404 - Not Found	When the device data is not found.
	500 - Internal Server Error	
	503 - Service Unavailable	DSR Connection Failure.
MONTE IP Troubleshooting API	404 - Not Found	When the device data is not found.
	500 - Internal Server Error	