Oracle® Communications
User Data Repository

Feature Configuration Guide

Release 12.11.0

F56868-01

May 2022

Oracle Communications User Data Repository Feature Configuration Guide, Release 12.11.0

F56868-01

This documentation is in preproduction status and is intended for demonstration and preliminary use only. It may not be specific to the hardware on which you are using the software. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to this documentation and will not be responsible for any loss, costs, or damages incurred due to the use of this documentation.

The information contained in this document is for informational sharing purposes only and should be considered in your capacity as a customer advisory board member or pursuant to your beta trial agreement only. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle Master Agreement, Oracle License and Services Agreement, Oracle PartnerNetwork Agreement, Oracle distribution agreement, or other license agreement which has been executed by you and Oracle and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

Contents

List of Tables

1

Introduction

This document defines the procedure that are executed to configure the Oracle Communications User Data Repository Custom entities, Pool Spanning Options (PSO), SNMP, and Ud Client feature on User Data Repository network.

Purpose and Scope

The Ud Client and Pool Spanning Option features are installed by default User Data Repository product. For to use these features, you need to enable and configure Ud Client and Pooling Spanning Options using the interface that has been introduced in the main menu of User Data Repository after you log in.

This document explains the procedures in detail to enable and configure Custom entities, SNMP, Ud Client, and Pool Spanning Options features.

Glossary

This section lists terms specific to this document. Table 2 Acronyms lists the acronyms used in the document.

Term	Definition
Auto-Enrollment	The ability for the SPR to create a Subscriber profile for an unrecognized subscriber identity, based on a pre-determined message received on one of the provisioning or traffic interfaces. The identity contained in the received message is used to create a default profile in the database.
Basic Pool	Refers to the existing quota pooling capabilities prior to this feature. Basic pools support up to 25 members.
Diameter Sh TPS	The number of Diameter transactions per second that are supported on the Sh signaling interface. A transaction is comprised of one Diameter Sh message received plus one message sent plus all of the processing required within the UDR system to handle the request
Enterprise Pool	A new type of pool introduced by this feature. This pool supports the sharing of pool quota across 1500 members.
Exhaustion	Exhaustion occurs when reports indicate that use of a metered unit has equaled or exceeded the specified quota limit. If a recurring Quota is exhausted, typically the sessions for the subscriber are subjected to more restrictive policies until the end of the Plan period or Billing Cycle.

Expiration

Expiration occurs when a periodic Quota reaches the end of the Plan period or Billing Cycle, or when a one-time quota reaches its established End Time or the close of its Validity period

Note: The time-based expiration of a Quota is different from the exhaustion of a Quota restricting the active session Time of the usage for the subscriber. A periodic Quota is typically Reset at expiration.

Non-Pool Host UDR

The UDR which hosts pool members for which pool data resides on Pool Host UDR.

Opaque Data

A data type that is incompletely defined in an interface, so that its values can only be manipulated by calling subroutines that have access to the missing information. The concrete representation of the type is hidden from its users.

Partial Pool Member

A pool member in a pool on the Non-Pool Host UDR in which the pool does not yet have the Pool Profile from the Pool Host UDR. This may be because the pool has not been created yet on the Pool Host UDR or because it was never received from the Pool Host UDR.

Whenever a partial pool member is added the Pool Entity data is requested from the Pool Host UDR.

Pass

A Pass is a one-time override which temporarily replaces or augments the default plan or service for the subscriber or service. While a Pass is in effect, it may modify the QoS controls, charging parameters, or other configurable rules associated with the service for a subscriber.

A Pass may:

- Be valid for a restricted interval
- Start when provisioned, or at a specific time, or upon the occurrence of a triggering event within its validity interval
- End at a specific time, or after given duration once activated, or upon a particular event
- Apply continuously, or only during certain time periods, or only under certain conditions (for example, when roaming)
- Apply to the overall usage for the subscriber, or be more limited (for example, applying only to specific applications, flows, traffic types, or pre-defined rules)

Passes are common options for pre-paid subscribers, who frequently have limited or no data access via their basic Plan, and may purchase Passes to gain access to such services. They can also be used to allow Casual Use plans for pre- or post-paid subscribers to purchase services on an occasional basis which they would not otherwise subscribe for on an ongoing basis.

Plan

The plan for a subscriber is the description of their basic, recurring service. Frequently, the Tier and/or Entitlement fields of profile data for the subscriber may be used to indicate or derive the plan type. Plans include enforceable policy characteristics (QoS and Charging parameters and PCC rules) computed automatically or through policy rules. A Plan may have associated Quota controls (see Basic Quota), which in turn may be subject to modification or over-ride through Passes, Top-ups, and Roll-overs (see below).

Pool Host UDR

The UDR which hosts pool data which may have pool members on other UDR systems.

Pool Network

Refers to the network of UDR servers across which a quota pool can span.

Provisioning TPS

Provisioning transactions per second, which is comprised of one provisioning message received plus one message sent plus all of the processing required within the UDR system to handle the transaction.

Quota

A Quota specifies restrictions on the amount of data Volume, active session Time, or service-specific Events that a subscriber can consume. A single Quota may express limits on any combination of Volume, Time, or Events. Quotas may be associated with a time period during which activity is measured.

Roll-over

Roll-over is a mechanism by which usage which was not consumed during one Quota period may be applied as a credit in a future period. Roll-over may apply to Basic Quotas associated with plan for a subscriber or may affect Passes or (more usually) Top-ups purchased by the subscriber. Roll-overs may be limited as to the amount that can be credited to the future period, or by capping the total amount of (basic and rolled-over) credit that may be available in a given period. They may also have limitations regarding the number of cycles that credits may be rolled into. In other words, Roll-over rules modify the process of resetting a recurring Quota.

Subscription Data Object

An SDO comprises of subscription state information combined with a collection of registers for storing entities. An SDO is accessed using an SDO ID and is stored in the UDR DB. SDOs come in two types: individual or pool. An individual SDO applies to one subscriber. A pool SDO applies to a group of subscribers.

Transparent Data

A data type whose representation is visible to the users.

Threshold

A Threshold is a soft limit at which usage must be reported during the monitoring of a Quota, usually lower than the full limit associated with the Quota. Typically, service parameters are not adjusted when a Threshold is reached, but other actions may be taken, such as notifying the user of their current usage.

Transparent Data

A data type whose representation is visible to the users.

Threshold

A Threshold is a soft limit at which usage must be reported during the monitoring of a Quota, usually lower than the full limit associated with the Quota. Typically, service parameters are not adjusted when a Threshold is reached, but other actions may be taken, such as notifying the user of their current usage.

Top-up

A Top-up is a modifier which takes effect only upon exhaustion of Basic Quota associated with plan or default service for a subscriber. Top-ups allow the subscriber to extend their access to services beyond the time or volume limits typically enforced.

Ud Client

An Ud Client is an FE that uses the Ud Interface to access subscriber data from a User Data Repository

Ud Interface

The Ud Interface is an access protocol as defined in 3GPP TS 29.335. It defines a logical connection between a Front-End (FE) and a User Data Repository.

The Ud Interface consists connections using LDAP to perform CRUD operations on subscriber data (Create/Delete/Update/Read), and connections using SOAP for publishing/subscribe interface in order to request notifications when subscriber data stored in the User Data Repository changes, and to receive those notifications when the data is changed.

Ud Server

An Ud Server is a User Data Repository that has an Ud Interface to allow external Front-Ends to access subscriber data via LDAP and SOAP, according to the Ud Interface specification.

Acronyms

This section explains the acronyms used in the document.

Table 1-1 Acronyms

Acronym	Definitions
3GPP	Third-Generation Partnership Project
AAA	Authorize-Authenticate-Answer
AAR	Authorize-Authenticate-Request
ADC	Application Detection and Control
AF	Application Function
AMBR	Aggregate Maximum Bit Rate
ARP	Allocation Retention Priority
AVP	Attribute Value Pair
BSS	Business Support System

Table 1-1 (Cont.) Acronyms

CALEA	Communications Assistance for Law Enforcement Act.
CCA	Credit-Control-Answer (CC-Answer)
CCR	Credit-Control-Request (CC-Request)
CMP	Configuration Management Platform
CSCF	Call Session Control Function
DCC	Diameter Credit Control
DPI	Deep Packet Inspection
DRA	Diameter Routing Agent
DSR	Diameter Signaling Router
FRS	Feature Requirements Specification
GBR	Guaranteed Bit Rate
G8, G9	Refers to the generation of HP server hardware.
GUI	Graphical User Interface
HA	High Availability
HSS	Home Subscriber Server
HTTP	Hypertext Transfer Protocol
HW	Hardware
IE	Internet Explorer
IMS	IP Multimedia Subsystem
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LI	Lawful Intercept

Table 1-1 (Cont.) Acronyms

LIMF	Lawful Intercept Mediation Function
LVM	Logical Volume Manager
MA	Management Agent
MCD	Media Component Description
MP	Message Processor
MPE	Oracle Multimedia Policy Engine
MPE-R	Oracle Multimedia Policy Engine – Routing Mode
MPE-S	Oracle Multimedia Policy Engine – Serving Mode
MRA	Oracle Multiprotocol Routing Agent
MS	Mediation Server
NFV-MANO	Network Function Virtualization Management and Orchestration
NFVO	Network Functions Virtualization Orchestrator
NOAM	Network OAM
NW-CMP	Network-Level Configuration Management Platform
OAM	Operations Administration Maintenance
OCS	Online Charging Service
OM	Operational Measurement
OSSI	Operation Support System Interface
PCC	Policy and Charging Control
PCD	Policy Connection Director
PCEF	Policy and Charging Enforcement Function (GGSN, PGW, DPI)
PCRF	Policy Control Resource Function (Oracle MPE)
P-CSCF	Proxy CSCF
PDN	Packet Data Network
PGW	Packet Data Network Gateway

Table 1-1 (Cont.) Acronyms

PNR	Push-Notification-Request
PUR	Profile-Update-Request
SEC	Subscriber Entity Configuration
QCI	QoS Class Identifier
QoS	Quality of Service
RAR	Re-Auth-Request (RA-Request) SUPL
REST	Representational State Transfer
ROB	Release of Bearer
S-CMP	Site-Level Configuration Management Platform
S-CSCF	Serving CSCF
SGW	Serving Gateway
Sh	Diameter Sh Interface
SMPP	Short Message Peer-to-Peer
SMS	Short Message Service
SNR	Subscribe-Notification-Request
SPR	Subscriber Profile Repository
STA	Session-Termination-Answer
STR	Session-Termination-Request
SRA	Successful Resource Allocation
TDF	Traffic Detection Function
TPS	Transactions Per Second
UD	Upgrade Director
UDR	User Data Repository
UE	User Equipment
UM	Upgrade Manager
UMCH	Usage Monitoring Congestion Handling
VIM	Virtual Infrastructure Manager
VM	Virtual Machine

Table 1-1 (Cont.) Acronyms

VNF	Virtual Network Function
VO	Verification Office
XML	Extensible Markup Language

2

Configuring Custom Entities

This section provides information to create and configure customer specific entities using *Subscriber Entity Configuration*.

New custom entities can be created similar to the existing entities except for Subscriber profile/Pool profile keys.

> **Note:** Only one subscriber and pool profile entity shall exist in the system that is available by default after the UDR is installed.

You can create the entities either:

- **Field Based** similar to Profile which shall contain a Base Field Set
- **Element Based** similar to Quota entity which shall contain a single Field Set within a Base Field Set and the Field Set shall, in turn, contain other fields.

The customer entities could be created as:

- Transparent or Opaque
- Pooled or Non-pooled

All the custom entities support existing operations on the provisioning, Sh, import, and export interfaces.

> **Note:** The xmlconverter is supported only for transparent entities only.

Creating Custom Entities

The **Entity** page is used to add, edit, and delete entities from the subscriber entity configuration (SEC). An entity is a type of subscriber data stored in XML formats, such as quota, state, or profile of a subscriber.

Two types of entities can be defined: opaque and transparent. Opaque entity elements cannot be individually read or edited. Transparent entity elements and values inside those elements can be read and edited. These values can be edited with SOAP/REST provisioning software. When defining a new transparent entity, the following must be defined (in the order listed) using options on the **Transparent Entity** menu before you can save the new entity:

1. Field Set (if using): Configuring Field Set
2. Base Field Set: Configuring Base Field Set
3. Definition: Adding a definition

Note: The UDR system comes pre-configured with the following eight entities; each is pre-defined as either opaque or transparent:

- DynamicQuota (opaque)
- PoolDynamicQuota (opaque)
- PoolProfile (transparent)
- PoolQuota (transparent)
- PoolState (opaque)
- Profile (transparent)
- Quota (transparent)
- State (opaque)

These pre-defined entities come preloaded with default configuration which can be customized if needed.

Additionally, an entity can be pooled with other entities for pool processing. An entity that is not pooled applies to one subscriber and an entity that is pooled can be shared by multiple subscribers.

Note: Once an entity is defined, at least one interface entity map entry must exist in order for the entity data to be accessed using interfaces such as provisioning, Sh etc. One interface entity map entry is required per interface.

Important! You cannot duplicate the subscriber profile. Custom entities with subscriber's unique ID (IMSI, MSISDN, Account ID, and NAI) and Pools unique ID (Pool Id) cannot be created.

Creating an Entity

This section details the procedure to create an entity.

To create an entity:

1. From the main menu, select **UDR**.
2. Select **Subscriber Entity Configuration**.
3. Select **Entity** .

   The **Entity** page appears. The list of entities and their entity type, along with their association with transparent entity definition, pooled, and registered ID is displayed.
4. Click **Insert**.

   The **Entity [Insert]** page appears.
5. Enter a unique **Entity Name** to associate with the entity.
6. Select the **Entity Type** from the pulldown menu. See Entity elements for more information on each field.

7. Check mark Pooled if this is a pooled entity.
8. Perform one of the following:

   - Click **OK** to save the permissions and return to the **Entity** page.

     Click **Apply** to save the permissions and remain on the **Insert** page.
   - Click **Cancel** to return to the **Entity** page without saving the changes.

   If an entity type of Transparent is selected, specify the type of **Transparent Entity Definition**.

   If **OK** or **Apply** is clicked and if any of the following exist, an error message appears:

   - Any required field is empty; no value was entered or selected
   - The entry in any field is not valid (wrong data type or out of the valid range)
   - The Entity Name is not unique; it already exists in the entity list

You have successfully created an entity.

Note: To change the order of the records in a column (from descending to ascending or vice versa), click the header of that column.

Note: Within Non-pooled entity or Pooled entity, no two entities should have same element string. Element string shall be unique for each entity within the pool or Non-Pool entity.

Entity elements

Use the Entity page to manage entities. This table describes fields on this page.

Table 2-1 Subscribers Elements

Element	Description	Data Input Notes
Entity Name	Name of the entity. Note that this value cannot be changed once a new entity is saved.	Format: String - valid characters are alphanumeric and underscore; must contain at least one alphabetic or numeric character, and cannot start with a number Range: 1-32 characters Default: Blank
Entity Type	Entity type can be opaque or transparent. When opaque is selected, the **Transparent Entity Definition** field is not accessible.	Format: pulldown list Range: Opaque, Transparent Default: Opaque

Table 2-1 (Cont.) Subscribers Elements

Element	Description	Data Input Notes
Transparent Entity Definition	Name of the transparent entity definition (this field associates a base field set and field set(s) (when used) with an entity). This field can be accessed only when the **Entity Type** is specified as Transparent.	Format: pulldown list Range: Select value from list Default: Blank
Pooled	Check box that indicates whether or not an entity is a pooled entity.	Format: Check box Range: Checked, unchecked Default: Unchecked
Register ID	Upon creation, every entity is automatically assigned a Unique register identity. This value is used internally, and cannot be edited.	Format: Numeric

Configuring Field Set

The **Base Field Set** page is used to add, edit, and delete base field sets from the subscriber entity configuration (SEC). A base field set is the first XML element (also called the root element) in an entity; it contains the entity's version information (if present in the entity) and one or more fields/field sets. There is only one base field set per entity.

The list of transparent entity field sets is displayed.

> **Note:** To change the order of the records in a column (from descending to ascending or vice versa), click the header of that column.

To create a Field Set:

1. From the main menu, select **UDR.**
2. Select **Subscriber Entity Configuration.**
3. Select **Transparent Entity.**
4. Select **Field Set**.

   The **Field Set** page appears.
5. Click **Insert**.
6. Enter a unique **Field Set Name** to associate with the entity.
7. Enter a unique **Element String** for the XML element name.
8. If this field set is repeatable, check mark **Repeatable** and enter an XML attribute instance identifier.

9. Click **Add Field** to add a data field record to associate with this transparent entity field set, and then enter data for the record. At least one data field record must be associated with a transparent entity field set or an error will occur when you try to save the field set. See, Field Set elements for more information about each variable element.

10. Once all needed fields have been added, perform one of the following:

    - Click **OK** to save the permissions and return to the **Field Set** page.
    - Click **Apply** to save the permissions and remain on the **Insert** page.
    - Click **Cancel** to return to the Field Set page without saving the changes.

    If **OK** or **Apply** is clicked and any of the following exist, an error message appears:

    - Any required field is empty; no value was entered or selected
    - The entry in any field is not valid (wrong data type or out of the valid range)
    - The field set name is not unique; it already exists in the field set list

You have successfully configured Field Set.

Field Set elements

Use the Field Set page to manage transparent entity field sets. Table 2-2 describes the elements that can be configured for a field set.

Table 2-2 Field Set Elements

Element	Description	Data Input Notes
Field Set Name	Name of the transparent entity field set.	Format: String - valid characters are alphanumeric and underscore; must contain at least one alphabetic or numeric character, and cannot start with a number Range: 1-64 characters Default: Blank
Element String	XML element name under which defined fields reside.	Format: String - valid characters are alphanumeric and underscore; must contain at least one alphabetic or numeric character, and cannot start with a number Range: 1-64 characters Default: Blank

Table 2-2 (Cont.) Field Set Elements

Element	Description	Data Input Notes
Repeatable	Indicates that the field set is a repeatable element. When multiple instances of data exist in an entity, a repeatable element contains the data fields for a specific instance of an entity. When this element is checked, the **Instance Identifier Attribute** element is enabled.	Format: Checkbox Range: Checked, unchecked Default: Unchecked
Instance Identifier Attribute	XML attribute instance associated with a repeatable element. This value indicates which repeatable element is being selected. This element is enabled only if the **Repeatable** element is checked.	Format: String - valid characters are alphanumeric and underscore; if enabled, must contain at least one alphabetic or numeric character, and cannot start with a number Range: 0-64 characters Default: Blank

In the lower part of the Field Set page, use the **Add Field** button to add a data field record to this field set. Table 2-3 describes the elements that can be configured for each data field record.

Table 2-3 Field Elements

Element	Description	Data Input Notes
Element String	The name of the XML element defined in the field set. If *Field Set* is selected from the **Type** dropdown list, then the **Element String** value for the field is automatically populated with the **Element String** value of the field set selected from the **Field Set Name** dropdown list.	Format: String; if the parent *Base Field Set* is Element Based, the format of **Element String** is alphanumeric and underscore, must contain at least one alphabetic or numeric character, and cannot start with a number Range: 1-64 characters Default: Blank

Table 2-3 (Cont.) Field Elements

Element	Description	Data Input Notes
Type	Field type. If *Field Set* is selected, then the **Min/Max Value**, **Updatable**, **Resettable**, **Reset Value**, **Defaultable**, **Default Value**, and **Special Format** elements are disabled. If *RegEx* is selected and **Resettable** or **Defaultable** is check marked, the value in the associated element (**Reset Value** or **Default Value**) is validated using the regex expression in the **Special Format** element.	Format: Dropdown list Range: Integer, Regex, Field Set Default: Regex
Field Set Name	Previously defined transparent entity field set where the field will be assigned. Note that a specific **Field Set Name** can only be referenced once by another field set or base field set. This element is enabled only if *Field Set* was selected from the **Type** dropdown list.	Format: Dropdown list Range: Previously defined tranparent entity field set Default: Blank
Min Value	The minimum field value allowed.	Format: Numeric Range: 0-4294967295 Default: Blank
Max Value	The maximum field value allowed.	Format: Numeric Range: 0-4294967295 Default: Blank
Updatable	Indicates that the value can be sent in an update request.	Format: Check box Range: Checked, unchecked Default: Unchecked
Resettable	Sets the value to the value in the **Reset Value** element if a reset operation is performed.	Format: Checkbox Range: Checked, unchecked Default: Unchecked

Table 2-3 (Cont.) Field Elements

Element	Description	Data Input Notes
Reset Value	Value to set the field to if a reset operation is performed. This element is enabled only if the **Resettable** element is checked. Note that if a **Type** of `RegEx` is specified, this value is validated using the regex expression specified in the **Special Format** element.	Format: Numeric Range: 0-4294967295 Default: Blank
Defaultable	Sets the value to the value in the **Default Value** element if a value if not specified in the insert request.	Format: Checkbox Range: Checked, unchecked Default: Unchecked
Default Value	Value to set the field to if a value is not specified in the insert request. This element is enabled only if the **Defaultable** element is checked. Note that if a **Type** of `RegEx` is specified, this value is validated using the regex expression specified in the **Special Format** element.	Format: Numeric Range: 0-4294967295 Default: Blank
Min Occur	Indicates if the field is mandatory or optional. If the value is 0, then the field does not have to exist. If the value is 1 or more, then the field is mandatory, and the specified number of instances must exist for the Entity to be valid. Note that when **Defaultable** is check marked, this value must be either 0 or 1.	Format: Numeric Range: 0-4294967295 Default: Blank
Max Occur	Indicates the maximum number of occurrences of the field. A value of *NO_LIMIT* indicates an unlimited number of occurrances.	Format: Numeric Range: 0-4294967295 Default: Blank

Table 2-3 (Cont.) Field Elements

Element	Description	Data Input Notes
Special Format	This element is a regular expression, conforming to the Perl regular expression syntax. This value must match the entire regular expression to be considered valid. This value can be used to apply advanced/special field validation if required (i.e., validation of the value in the **Default Value** or **Reset Value** element).	Format: String Range: 1-64 characters Default: Blank

Configuring Base Field Sets

To create a Base Field Set:

1. From the main menu, select **UDR**.
2. Select **Subscriber Entity Configuration**.
3. Select **Transparent Entity**.
4. Select **Base Field Set**.

   The **Base Field Set** page appears. The list of transparent entity base field sets is displayed.

   Note: To change the order of the records in a column (from descending to ascending or vice versa), click the header of that column.

5. Click **Insert**.
6. Enter a unique **Base Field Set Name** to associate with the entity.
7. Enter a unique **Element String** for the XML element name.
8. If multiple versions of this entity are allowed, check mark **Allow Versions** and enter a **Version Identifier Element** and **Version Value**.
9. In **XML Storage Format**, specify if the entity version is element based or field name based.
10. Click **Add** field to add a data field record to associate with this transparent entity base field set, and enter data for the record. At least one data field record must be associated with a transparent entity base field set or an error will occur when you try to save the base field set. See Base Field Set Elements for more information on each variable element.
11. Once all needed fields have been added, perform one of the following:

- Click **OK** to save the permissions and return to the **Base Field Set** page.
- Click **Apply** to save the permissions and remain on the **Insert** page.
- Click **Cancel** to return to the **Base Field Set** page without saving the changes.

If **OK** or **Apply** is clicked and any of the following exist, an error message appears:

- Any required field is empty; no value was entered or selected
- The entry in any field is not valid (wrong data type or out of the valid range)
- The base field set name is not unique; it already exists in the base field set list

Base Field Set elements

Use the Base Field Set page to manage base field sets. Table 2-4 describes fields on this page.

Table 2-4 Base Field Set Elements

Element	Description	Data Input Notes
Base Field Set Name	Name of this Base Field Set Definition	Format: String - alphanumeric and underscore; must contain at least one letter or number, and cannot start with a number Range: 1-64 characters Default: Blank
Element String	XML element name under which defined fields reside.	Format: String - alphanumeric and underscore; must contain at least one letter or number, and cannot start with a number Range: 1-64 characters Default: Blank
Allow Versions	Indicates whether multiple versions of this entity are allowed.	Format: Checkbox Range: Checked, unchecked Default: Unchecked
Version Identifier Element	The XML element name of the version identifier. This element is enabled only if the **Allow Versions** element is checked.	Format: String - alphanumeric and underscore; must contain at least one letter or number, and cannot start with a number Range: 1-64 characters Default: Blank

Table 2-4 (Cont.) Base Field Set Elements

Element	Description	Data Input Notes
Version Value	The version value of the XML blob associated with the base field set. This element is enabled only if the **Allow Versions** element is checked.	Format: Numeric Range: 0-4294967295 Default: Blank
XML Storage Format	Indicates whether the entity version is element based or field name based.	Format: Dropdown list Range: Element Based, Field Name Based Default: Element Based

In the lower part of the Base Field Set page, use the **Add Field** button to add a data field record to the base field set. Table 2-5 describes the elements that can be configured for each data field record.

Table 2-5 Base Field Elements

Element	Description	Data Input Notes
Element String	XML element name under which defined fields reside. If *Field Set* is selected from the **Type** dropdown list, then this **Element String** value is automatically populated with the **Element String** value of the field set selected from the **Field Set Name** dropdown list.	Format: String - alphanumeric only; must contain at least one letter or number. String value can start with a number if **XML Storage Format** is *Field Name Based*, value cannot start with a number if **XML Storage Format** is *Element Based*. Range: 1-64 characters Default: Blank
Type	Field type. If *Field Set* is selected, then the **Min/Max Value**, **Updatable**, **Resettable**, **Reset Value**, **Defaultable**, **Default Value**, and **Special Format** elements are disabled. If *RegEx* is selected and **Resettable** or **Defaultable** is check marked, the value in the associated element (**Reset Value** or **Default Value**) is validated using the regex expression in the **Special Format** element.	Format: Dropdown list Range: Integer, Regex, Field Set Default: Regex

Table 2-5 (Cont.) Base Field Elements

Element	Description	Data Input Notes
Field Set Name	Previously defined transparent entity field set where the field will be assigned. Note that a specific **Field Set Name** can be referenced only once by another field set or base field set. This element is enabled only if *Field Set* was selected from the **Type** dropdown list.	Format: Dropdown list Range: Previously defined transparent entity field set Default: Blank
Min Value	The minimum field value allowed.	Format: Numeric Range: 0-4294967295 Default: Blank
Max Value	The maximum field value allowed.	Format: Numeric Range: 0-4294967295 Default: Blank
Updatable	Indicates that the field value can be sent in an update request.	Format: Checkbox Range: Checked, unchecked Default: Unchecked
Resettable	Sets the field value to the value in the **Reset Value** element if a reset operation is performed.	Format: Checkbox Range: Checked, unchecked Default: Unchecked
Reset Value	Value to set the field if a reset operation is performed. This element is enabled only if the **Resettable** element is checked. Note that this value is validated using the regex expression specified in the **Special Format** element.	Format: Numeric Range: 0-4294967295 Default: Blank
Defaultable	Sets the field value to the value in the **Default Value** element if a value if not specified in the insert request.	Format: Checkbox Range: Checked, unchecked Default: Unchecked

Table 2-5 (Cont.) Base Field Elements

Element	Description	Data Input Notes
Default Value	Value to set the field if a value is not specified in the insert request. This element is enabled only if the **Defaultable** element is checked. Note that this value is validated using the regex expression specified in the **Special Format** element.	Format: Numeric Range: 0-4294967295 Default: Blank
Min Occur	Indicates if the field is mandatory or optional. If set to 0, then the field does not have to exist. If set to 1 or more, then the field is mandatory, and the specified number of instances must exist for the Entity to be valid.	Format: Numeric Range: 0-4294967295 Default: Blank
Max Occur	Indicates the maximum number of occurrences of the field. A value of *NO_LIMIT* indicates an unlimited number of occurrences.	Format: Numeric Range: 0-4294967295 Default: Blank
Special Format	This element is a regular expression conforming to the Perl regular expression syntax. This value must match the entire regular expression to be considered valid. This value can be used to apply advanced/special field validation if required (i.e., validation of the value in the **Default Value** or **Reset Value** element.) .	Format: String Range: 1-64 characters Default: Blank

Adding a Definition

The **Definition** page is used to add, edit, and delete transparent entity definitions from the subscriber entity configuration (SEC). A transparent entity definition is a container for different versions of transparent entities and provides details on if/how to extract the version number from within a transparent entity.

To create a definition:

1. From the main menu, select **UDR.**
2. Select **Subscriber Entity Configuration.**

3. Select **Transparent Entity.**
4. Select **Definition**.
5. Enter the text of the second step here.

   The **Definition** page appears.
6. Click **Insert**.
7. Enter a unique **Transparent Entity Definition Name** to associate with the definition.
8. Checkmark **Allow Reset** if a reset can be performed for this entity definition.
9. Click **Add Version** for each base field set to associate with this entity definition.
10. From the **Base Field Set Name** dropdown list, select an existing base field set to associate with this entity definition. Note that at least one base field set must be associated with a definition before you can save the new definition record.

    > **Note:** The base field set info that appears here is for display purposes only. See Base Field Set elements for more information on each field.
11. Once all needed fields have been added, perform one of the following:
    - Click **Apply** to save the permissions and remain on the **Insert** page.
    - Click **OK** to save the permissions and return to the **Definition** page.
    - Click **Cancel** to return to the **Definition** page without saving the changes.

    If **OK** or **Apply** is clicked and any of the following exist, an error message appears:
    - Any required field is empty; no value was entered or selected
    - The entry in any field is not valid (wrong data type or out of the valid range)
    - The transparent entity definition name is not unique; it already exists in the base field set list

You have successfully added a definition.

> **Note:** A transparent entity definition can be defined but cannot be accessed until it is associated with an entity. This can be done by either creating a new entity to use the definition or updating the definition associated with an existing entity. An opaque entity must be changed to transparent before it can be associated with a definition.

Adding Interface Entity Maps

The **Interface Entity Maps** menu option maps the name of an entity in the SEC database to a specific type of interface - SH, REST, SOAP, etc.

An interface entity mapping makes an entity available on the related interface. For example, in the case of a REST interface, the data type seen within the URL of a REST request is mapped to the entity. The example uses of interface entity maps:

Sh Interface Entity Map - mapping a data entity to the Sh interface makes the entity available to the Policy Management server.

REST Interface Entity Map - mapping a data entity to the REST interface makes the entity available to the provisioning systems

To add an interface entity map:

1. 1. From the main menu, select **UDR.**
2. 2. Select **Subscriber Entity Configuration**.
3. 3. Select **Interface Entity Map**.

   The **Interface Entity Map** page appears. The list of existing interface entity maps is displayed.

   > **Note:** To change the order of the records in a column (from descending to ascending or vice versa), click the header of that column.

4. Click **Insert**.
5. Enter an **Interface Entity Name** to map to a specific type of interface and entity. If needed, the same **Interface Entity Name** can be used for more than one Interface Name. See Interface Entity Map elements for more information on each field.
6. Select a type of interface from the **Interface Name** drop-down list.
7. Select an existing entity name to associate with this interface from the **Entity Name** pulldown list.
8. Perform one of the following:
   - Click **OK** to save the permissions and return to the **Interface Entity Map** page.
   - Click **Apply** to save the permissions and remain on the **Insert** page.
   - Click **Cancel** to return to the **Interface Entity Map** page without saving the changes.

   If **OK** or **Apply** is clicked and any of the following exist, an error message appears:

   - Any required field is empty; no value was entered or selected
   - The entry in any field is not valid (wrong data type or out of the valid range)
   - The interface entity name is not unique; it already exists in the interface entity map list

Interface Entity Map elements

Use the Interface Entity Map page to manage interface entity maps. This table describes fields on this page.

Table 2-6 Interface Entity Map Elements

Element	Description	Data Input Notes
Interface Entity Name	Entity name specific to an interface. For example, Service Indication name (for Sh) and URL token (for REST).	Format: String - valid characters are alphanumeric and underscore; must contain at least one alphabetic or numeric character, and cannot start with a number Range: 1-128 characters Default: Blank
Interface Name	Type of interface	Format: Dropdown list Range: SH, REST, SOAP, XML Default: REST
Entity Name	SEC entity name to be mapped.	Format: Dropdown list Range: All defined entity names Default: Blank

3

Configuring Services for Dual Path HA

This section provides the procedure for updating Oracle Communications User Data Repository Services for the Dual Path HA feature. This procedure applies to all configurations that make use of a Secondary/DR Site.

Note: You must perform this configuration after completing the UDR Software Upgrade.

Configuring Services for Dual Path HA

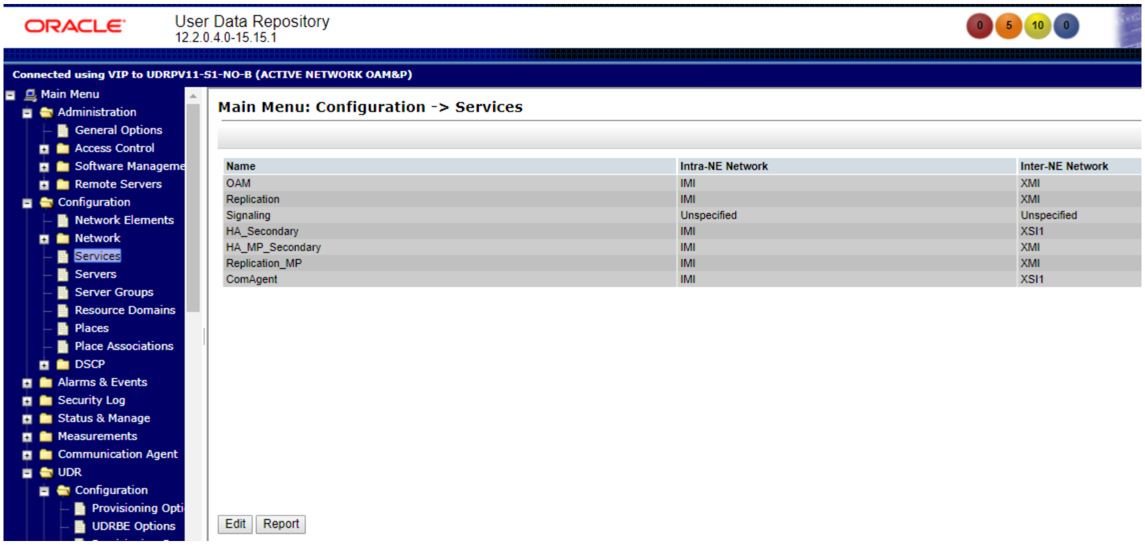
This section provides the procedure for updating Oracle Communications User Data Repository Services for the Dual Path HA feature. This procedure applies to all configurations that make use of a Secondary/DR Site.

Note: You must perform this configuration after completing the UDR Software Upgrade.

To configure services for dual path high availability:

1. Login to UDR.

2. From the main menu, select **Configuration** and then select **Services**.



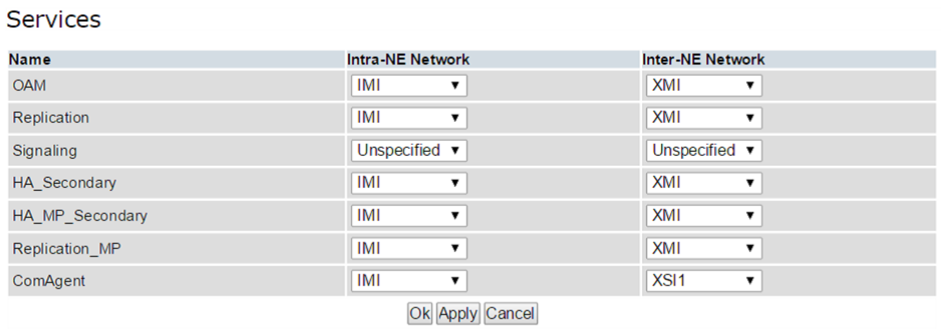
Connected using VIP to UDRPV11-S1-NO-B (ACTIVE NETWORK OAM&P)

Main Menu: Configuration -> Services

Name	Intra-NE Network	Inter-NE Network
OAM	IMI	XMI
Replication	IMI	XMI
Signaling	Unspecified	Unspecified
HA_Secondary	IMI	XSI1
HA_MP_Secondary	IMI	XMI
Replication_MP	IMI	XMI
ComAgent	IMI	XSI1

3. Click **Edit** and perform the following:

   a. For **HA_Secondary** service, in the Inter-NE HA_Secondary column, select **XMI** from the drop-down menu.



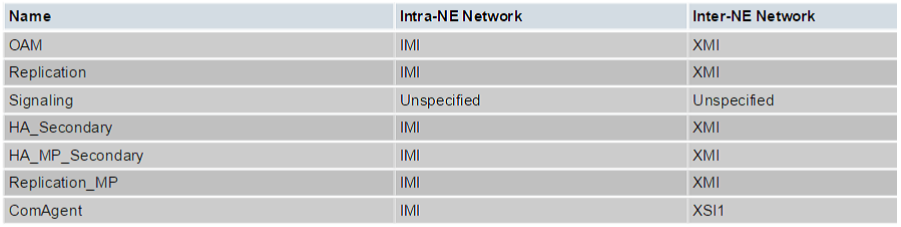
Services

Name	Intra-NE Network	Inter-NE Network
OAM	IMI ▾	XMI ▾
Replication	IMI ▾	XMI ▾
Signaling	Unspecified ▾	Unspecified ▾
HA_Secondary	IMI ▾	XMI ▾
HA_MP_Secondary	IMI ▾	XMI ▾
Replication_MP	IMI ▾	XMI ▾
ComAgent	IMI ▾	XSI1 ▾

   b. Click **Apply**.

   c. Click **OK** in the popup window.

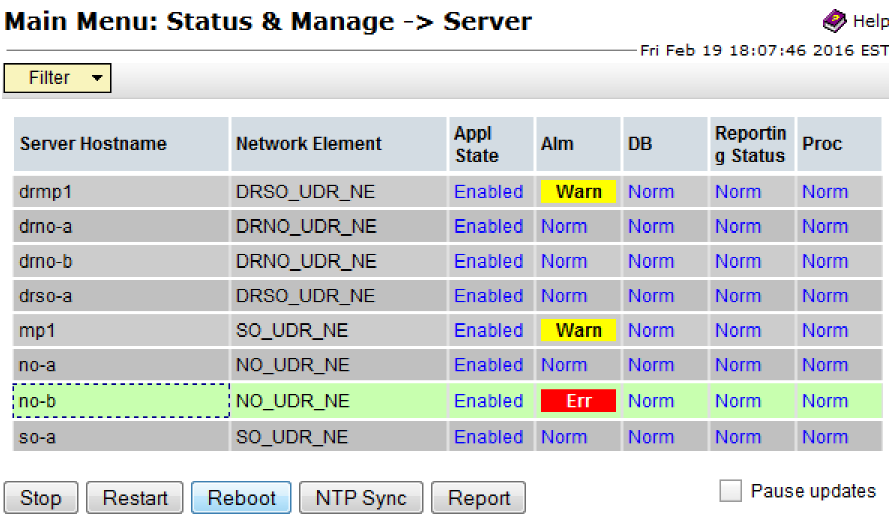
   NOAMP and MP Servers need to be restarted. The following Services Configuration screen appears.



Name	Intra-NE Network	Inter-NE Network
OAM	IMI	XMI
Replication	IMI	XMI
Signaling	Unspecified	Unspecified
HA_Secondary	IMI	XMI
HA_MP_Secondary	IMI	XMI
Replication_MP	IMI	XMI
ComAgent	IMI	XSI1

4. Restart/Reboot all NOAMP and MP Servers by doing one of the following:

   - From the main menu, select **Status & Manage** and then **Server**. The Server screen appears. Click **Reboot**.



Main Menu: Status & Manage -> Server

Fri Feb 19 18:07:46 2016 EST

Server Hostname	Network Element	Appl State	Alm	DB	Reporting Status	Proc
drmp1	DRSO_UDR_NE	Enabled	Warn	Norm	Norm	Norm
drno-a	DRNO_UDR_NE	Enabled	Norm	Norm	Norm	Norm
drno-b	DRNO_UDR_NE	Enabled	Norm	Norm	Norm	Norm
drso-a	DRSO_UDR_NE	Enabled	Norm	Norm	Norm	Norm
mp1	SO_UDR_NE	Enabled	Warn	Norm	Norm	Norm
no-a	NO_UDR_NE	Enabled	Norm	Norm	Norm	Norm
no-b	NO_UDR_NE	Enabled	Err	Norm	Norm	Norm
so-a	SO_UDR_NE	Enabled	Norm	Norm	Norm	Norm

- Run the following reboot command on the terminal of each server:`$ sudo reboot`

Note: You must execute the reboot command on all NOAMPs and MPs.

You have successfully configured services for dual path HA.

4

Pool Spanning Options Feature Configuration

This chapter describes the procedure to configure Pool Spanning Option for User Data Repository product.

Overview

Pool Spanning Option feature allows a subscriber pool quota to be shared by subscribers that are provisioned on separate User Data Repository systems. Whenever multiple User Data Repository systems are deployed within a network, subscribers are partitioned across the systems based on either IMSI or MSISDN ranges.

Pools can be defined on each of the User Data Repository systems. You can have pool quota services that include subscribers from different geographic regions, or that might have different devices that don't all fall within the IMSI ranges associated with a single OCUDR. This feature is intended to address those limitations.

Understanding Pool Spanning

Each User Data Repository is configured with a pool network that contains a list of other User Data Repository across which pools can span. This configuration data will include an identifying tag for each OCUDR, along with other details needed in order to be able to signal information between the systems. The OCUDR GUI will support the ability to define the ranges of **PoolIDs** that are associated with each UDR in the network. This data will be used in determining whether the UDR is a Pool Host UDR or Non-Pool Host UDR whenever a new pool is created.

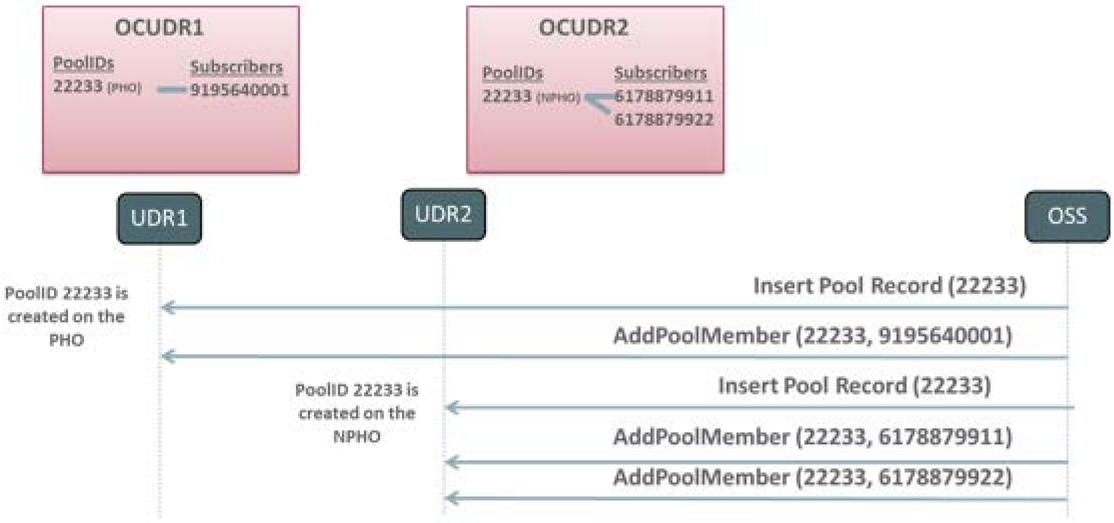
An insert request is used to create the pool, followed by **AddPoolMember**/ **DelPoolMember** requests to add or remove individual subscribers from the pool.

If a network is comprised of more than one User Data Repository, subscribers are partitioned between the User Data Repository based on either IMSI or MSISDN range. If a shared quota pool is intended to include subscribers from more than one User Data Repository, then the provisioning OSS shall generate an insert request to create the pool on each User Data Repository that contains pool subscribers. When processing the insert request, UDR applies the configuration data to determine whether it is the Pool Host UDR (PHO) or the Non-Pool Host UDR (NPHO) for the pool. One of the following results will occur:

- If the User Data Repository is the PHO, then the pool profile is created as it normally is, including any fields specified for the pool profile. The OSS may associate entity data as it normally would with the pool profile on the PHO, including PoolQuota, PoolState, and Pool DynamicQuota.

- If the UDR is the NPHO, then the pool profile is created, including any fields specified for the pool profile. However, entity data may not be associated with the pool profile on the NPHO. If any entity data is provided by the OSS, then the entity data is ignored.

For Example, See the Figure 1. In this example, two UDRs are in the pool network, and identifying information for each is configured on each UDR. This information stipulates that poolIDs starting with 2 are hosted on UDR1, and poolIDs starting with 3 are hosted on UDR2. The subscribers are partitioned based on MSISDN, with 919 subscribers on UDR1 and 617 subscribers on UDR2. A PSO 22233 is created, with UDR1 as the PHO and UDR2 as the NPHO. The respective subscribers from each UDR are added to the pool. The message sequence chart outlines the provisioning requests that are created by the OSS in order to create the Pool Spanning and how subscribers from each of the UDRs in the pool network are associated with it.


OCUDR1
PoolIDs
22233 (PHO)
Subscribers
9195640001
OCUDR2
PoolIDs
22233 (NPHO)
Subscribers
6178879911
6178879922
UDR1
UDR2
OSS
PoolID 22233 is created on the PHO
Insert Pool Record (22233)
AddPoolMember (22233, 9195640001)
PoolID 22233 is created on the NPHO
Insert Pool Record (22233)
AddPoolMember (22233, 6178879911)
AddPoolMember (22233, 6178879922)

Pool quota management only occurs if the pool has been properly configured on the PHO. Whenever the AddPoolMember request is used to add a subscriber to a pool on the NPHO, data for the subscriber is internally updated to identify the pool to which it belongs. Whenever that particular member (say 6178879911 in the figure above) creates a new session, the UDR hosting the subscriber (say UDR2 in the Figure 1) interacts with the UDR hosting the pool (UDR1 in Figure 1) to ensure that the pool exists on UDR1 and to be sure that any updates to pool quota, state, or dynamic quota on UDR1 are provided to UDR2 for the UDR2 pool members that have active sessions. If any data discrepancies are detected in the data between the UDRs (perhaps the pool has not been created on UDR1), then UDR2 manages the subscriber quota and ignores the association to the pool.

The following additional pool provisioning requests are available for Pool Spanning:

- DelPoolMember

  This request removes the subscriber or list of subscribers from the specified pool. This request must be sent to the UDR that hosts the subscriber (in the same way that AddPoolMember requests must be sent to the UDR that hosts the pool). Existing processing logic applies to scenarios where either the pool or one or more subscribers does not exist on the UDR that receives the request.

- GetPoolID

  This request contains the identity of a subscriber, and returns the poolID (if any) associated with the subscriber. This request must be sent to the UDR that hosts the subscriber profile. If the specified subscriber does not exist on the UDR that receives the GetPoolID request, then an error is returned.

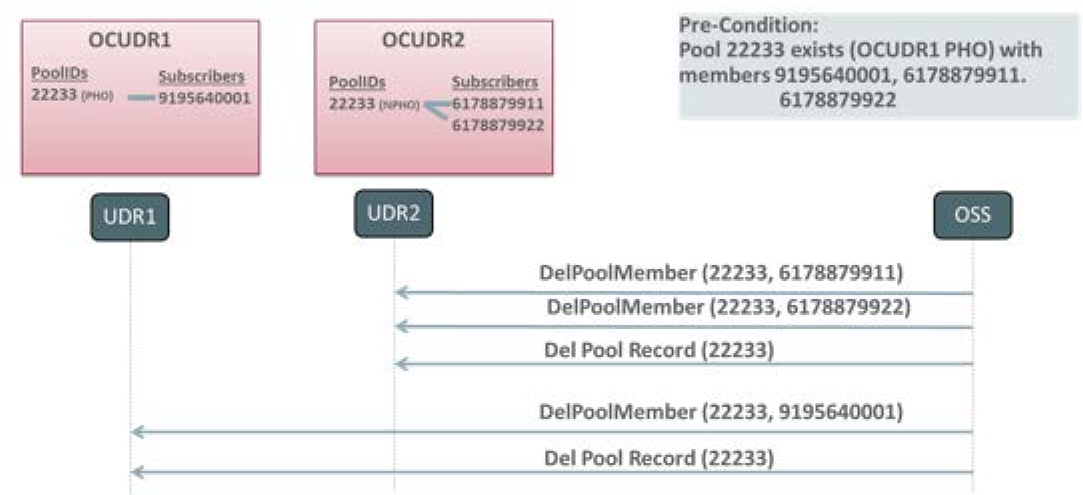
- GetPoolMembers

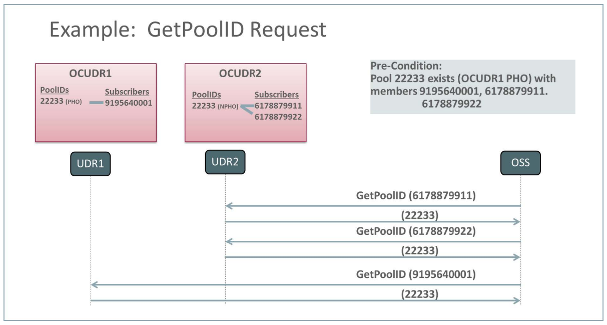
  The request is processed just as it currently is. Whenever a UDR receives this request, it returns all members hosted by the UDR that are associated with the specified pool. It does not include any PSO members that are hosted on other UDRs in the pool network.

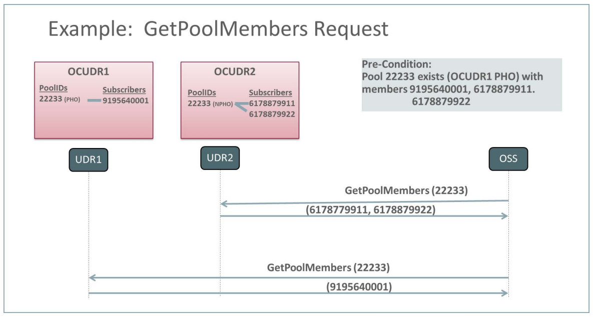
- GetAllPoolMembers

  This is a new provisioning request that can be generated by the OSS in order to get a complete list of all members associated with a pool, including any members from other UDRs in the pool network if the pool happens to be a PSO. A GetAllPoolMembers request to get all pool members across all UDR instances can be sent to any UDR in the pool network, and provides the same result regardless of if it is processed by the PHO or NPHO. If a GetAllPoolMembers request is received for a normal pool that does not span UDRs, provides results that are consistent with the GetPoolMembers request.

Figure 2, Figure 3, Figure 4, and Figure 5 provide examples of each of these commands. These examples assume the successful creation of the pool shown in Figure 1.

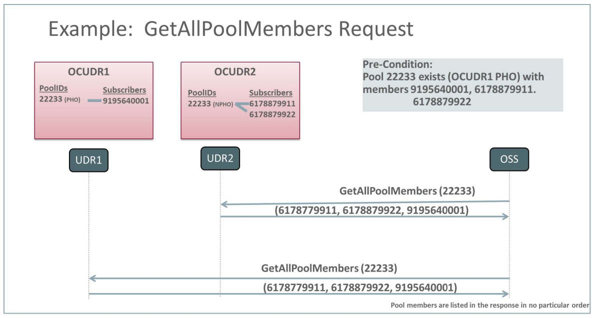

OCUDR1
PoolIDs
Subscribers
22233 (PHO)
9195640001
OCUDR2
PoolIDs
Subscribers
22233 (NPHO)
6178879911
6178879922
Pre-Condition:
Pool 22233 exists (OCUDR1 PHO) with members 9195640001, 6178879911. 6178879922
UDR1
UDR2
OSS
DelPoolMember (22233, 6178879911)
DelPoolMember (22233, 6178879922)
Del Pool Record (22233)
DelPoolMember (22233, 9195640001)
Del Pool Record (22233)





Example: GetPoolMembers Request

Pre-Condition:
Pool 22233 exists (OCUDR1 PHO) with members 9195640001, 6178879911.
6178879922



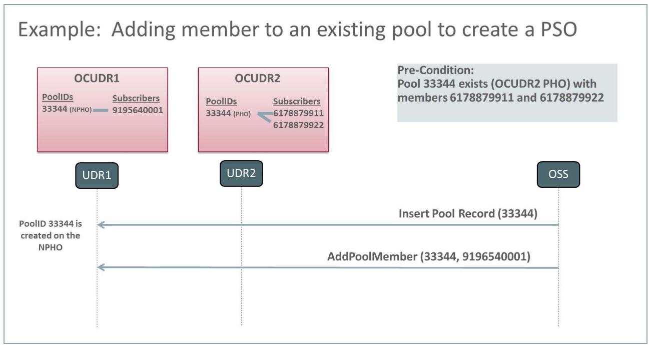
Example: GetAllPoolMembers Request

Pre-Condition:
Pool 22233 exists (OCUDR1 PHO) with members 9195640001, 6178879911.
6178879922

Pool members are listed in the response in no particular order

Pool Spanning does not alter the existing behavior associated with pools when all members associated with the pool are on the same UDR as the pool. It is possible, though, to convert an existing pool to a Pool Spanning by creating a pool profile on the NPHO, and adding new members to the pool on the NPHO. In the following example, assume PoolID 33344 already exists on UDR2 and contains subscribers 6178879911 and 6178879922.

With the introduction of this feature, the pool network is configured on both UDR1 and UDR2. If you want to add UDR1 subscriber 9195640001 to the pool, this can be accomplished by creating a pool profile for the Pool Spanning on UDR1, followed by an AddPoolMember request to UDR1 in order to add subscriber 9195640001 to the pool.



Example: Adding member to an existing pool to create a PSO

Pre-Condition:
Pool 33344 exists (OCUDR2 PHO) with members 6178879911 and 6178879922

PoolID 33344 is created on the NPHO

Following the completion of the upgrade to the release that contains this feature, each UDR within the pool network is configured with the identities of the other UDRs in the pool network, along with the PoolID ranges that are associated with the other UDRs in the pool network. All UDRs in the pool network must be upgraded to the release that contains the Pool Spanning feature before any Pool Spanning are provisioned. Although the validation of this feature is limited to a pool network of two

UDRs in the initial release, the implementation does not limit the ability to expand the pool network in the future to support additional UDRs.

- UDR request is received

  The UDR returns all subscriber data entities requested for the subscriber, including pool data entities if the subscriber has been associated with a pool. This provides consistent behavior with existing Sh interactions and is done transparently without regard to the UDR that hosts the pool that is associated with the subscriber.

- SNR (subscribe) request is received

  The UDR subscribes to all ServiceIndications for the subscriber, including pool service indications if applicable. This is done transparently without regard to the UDR that hosts the pool that is associated with the subscriber. If the subscriber data is requested as a part of the SNR, then the UDR returns all subscriber data entities requested for the subscriber, including pool data entities if the subscriber has been associated with a pool. This provides consistent behavior with existing Sh interactions and is done transparently without regard to the UDR that hosts the pool that is associated with the subscriber.

- PUR request is received

  The UDR updates the subscriber and pool data entities provided in the request. This is done transparently without regard to the UDR that hosts the pool that is associated with the subscriber. When the PUR is processed, data associated with the pool profile and associated entities are processed first, followed by data associated with the subscriber profile and associated entities. If some entities are not successfully updated, then a diameter UNABLE_TO_COMPLY response is delivered to the PCRF.

- SNR (un-subscribe) request is received

  The UDR unsubscribes from all ServiceIndications for the subscriber, including pool service indications if applicable. If the subscriber data is to be included in the SNR response, then the UDR returns all subscriber data entities requested for the subscriber, including pool data entities if the subscriber has been associated with a pool. This provides consistent behavior with existing Sh interactions and is done transparently without regard to the UDR that hosts the pool that is associated with the subscriber.

- PNR request is generated (due to updated data)

  The UDR generates a PNR for each subscriber that has an active subscription for notifications if any entity associated with that subscriber is updated. This applies to both subscriber entities (profile, Quota, and State), as well as pool entities (profile, PoolQuota, PoolState). In some Pool Spanning scenarios, the pool data and subscriber data may be hosted on different UDRs. In this case, if both pool and subscriber data are updated by a single request or <tx> transaction, then the resulting number of PNR messages is based on whether NotifEff is set.

  – If NotifEff is set, then a single PNR is generated for the subscriber that includes both the pool and subscriber updates.

  – If NotifEff is not set, then a separate PNR is generated for each updated entity.

Table 4-1 Pool Spanning Configuration Insert Window Fields

Field	Description
UDR Name	Indicates a Unique Name Identifier for the User Data Repository, which is a case-insensitive string. The default value is n/a. you can enter a maximum of 15 character string. **Note:** Valid characters are alphanumeric and underscore. **Note:** . ***Important: The string must contain at least one alpha and must not start with a digit. It is a mandatory field.***
UDR ID	Indicates a non-zero and, unique UDR Instance ID. The default value is n/a. Range is 1to 4294967295.] [A value is required.]
Type	The flag indicates if UDR ID is Host or Remote. Self for Host UDR and Remote for Remote UDR. This is set to Self only for Host UDR. By default, the value is remote.

Permissions to Access Pool Spanning Options

You can grant permission for a user to access Pool Spanning. The Administrator can pool spanning by default.

To give users access to Pool Spanning feature::

1. Login to User Data Repository main menu.
2. Click **Administration**.
3. Click **Access Control**.
4. Click **Groups**.
5. Click **Insert**.
6. Enable the following permissions for the user as required to View, Insert, Edit, Delete, and Manage.
   - Pool Spanning Options
   - Pool Network Configuration
   - UDR Key Range

You have successfully given the permissions to the selected group. The administrative group, admin have access to all the permissions by default.

Configuring Pool Spanning

This section explains the procedure to configure Pool Spanning feature.

To configure Pool Spanning functionality:

1. Install and configure User Data Repository product on multiple machines. For example, UDR1 and UDR2.
2. Configure ComAgent on OCUDR1 and OCUDR2 as described in the Configuring ComAgent for UDR Machines.
3. Configure Pool Network Table on UDR1 and UDR2. See, Configuring Pool Network.
4. Add entries in the UDR Key Range table on UDR1 and UDR2 for the pool id ranges. See, Inserting UDR Key Range.
5. Optional) Execute the following command on Active NO as a root user only for systems upgraded from User Data Repository product release 12.1 to User Data Repository product release 12.11.0.

   This step is not required for a system upgraded from User Data Repository product release10.2 or a new installation of User Data Repository product release 12.11.0.

   ```
   iset -fflags=0 Subscription where "flags!=0"
   ```

   The command may take up to three hours to complete for a 30M Database. Proceed to step 6 only after completion of Step 5.
6. Run the following command to activate Pool Spanning on Active NO as a root user on User Data Repository system product.

   ```
   iset -fvalue=TRUE CommonOptions where "var='PSO_Enabled'"
   iset -fvalue=TRUE CommonOptions where
   "var='PoolProfileMergeEnabled'"
   ```

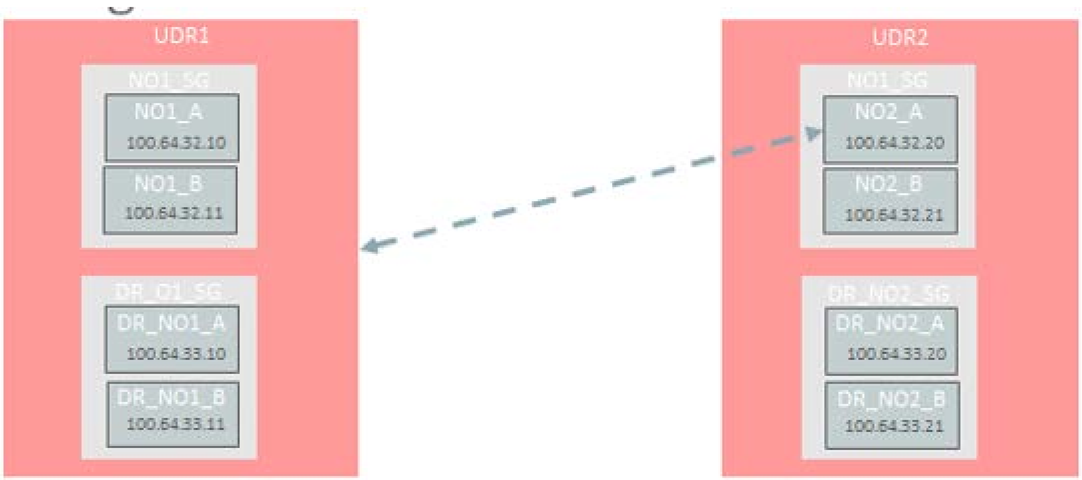
You have successfully configured the Pool Spanning Feature.

Configuring ComAgent for UDR Machines

This section explains the procedure to configure ComAgent for UDR machines.

To configure ComAgent for UDR machines:

1. Configure UDR1 as a high availability service provider for each NO in UDR2.



2. From Main Menu, click Communication Agent.
3. Click Configuration.
4. Click Remote Servers.
5. Click Insert.

   a. In the Remote Server Name field, enter unique identifier used to label a Remote Server.

      Note: You can enter a maximum of 32 character string. The valid characters are alphanumeric, minus sign, period and underscore. The string must start with an alphanumeric or an underscore and end with an alphanumeric.

   b. In the Remote Server IPv4 IP Address field, enter the IPv4 IP address of the Remote Server. If IPv6 IP is specified then IPv4 IP address is optional.

   c. From the Remote Server Mode list, select the server mode. The available options are Client and Server.

   d. From the IP Address Preference list, select IP address preference for connection establishment. The available options are:

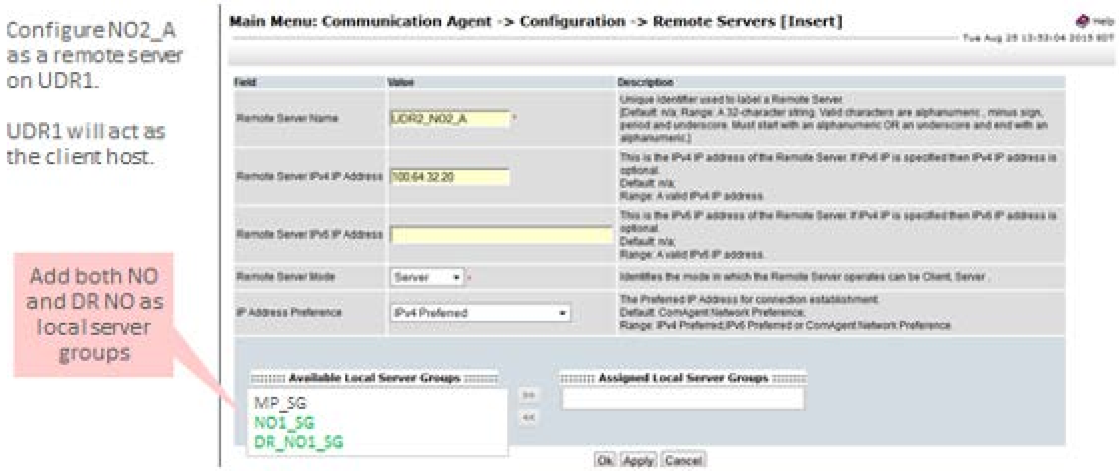
   e. In the assigned Local Server Groups field, add the server groups which can be associated with the Remote Server. The Servers in these server groups establish connections with this Remote Server. Server Groups which are available in the Available Local Server Groups list and the servers associated with the remote server are in the Assigned Local Server Groups list.

   f. Click **OK** to insert the remote server.

   g. Click **Apply** to configure the remote server.

   - ComAgent Network Preference
   - IPv4 Preferred
   - IPv6 Preferred

   The Insert window opens.



Configure NO2_A as a remote server on UDR1.

UDR1 will act as the client host.

Add both NO and DR NO as local server groups

Main Menu: Communication Agent -> Configuration -> Remote Servers [Insert]

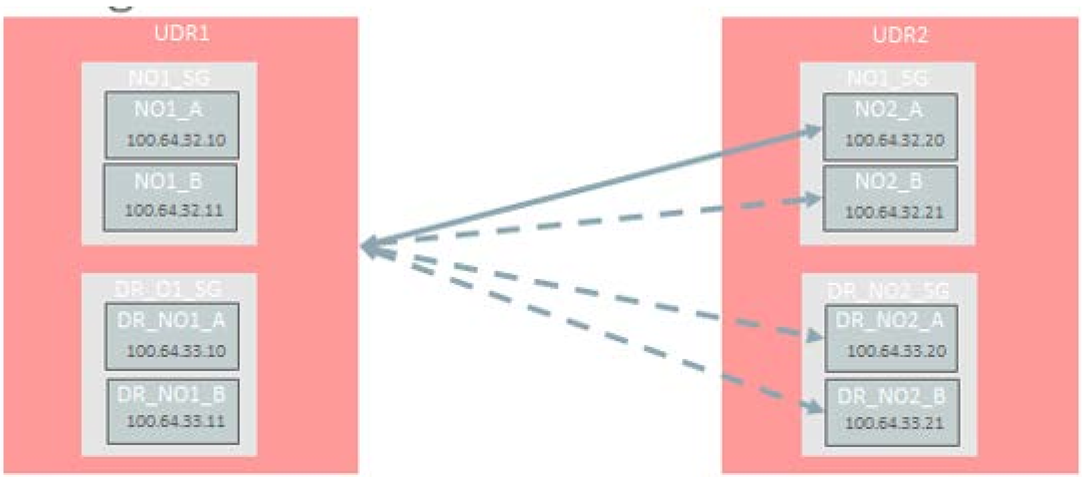
Tue Aug 25 13:53:04 2015 EDT

Field	Value	Description
Remote Server Name	UDR2_NO2_A *	Unique identifier used to label a Remote Server. [Default: n/a; Range: A 32-character string. Valid characters are alphanumeric , minus sign, period and underscore. Must start with an alphanumeric OR an underscore and end with an alphanumeric.]
Remote Server IPv4 IP Address	100.64.32.20	This is the IPv4 IP address of the Remote Server. If IPv6 IP is specified then IPv4 IP address is optional. Default: n/a; Range: A valid IPv4 IP address.
Remote Server IPv6 IP Address		This is the IPv6 IP address of the Remote Server. If IPv4 IP is specified then IPv6 IP address is optional. Default: n/a; Range: A valid IPv6 IP address.
Remote Server Mode	Server *	Identifies the mode in which the Remote Server operates can be Client, Server .
IP Address Preference	IPv4 Preferred	The Preferred IP Address for connection establishment. Default: ComAgent Network Preference. Range: IPv4 Preferred,IPv6 Preferred or ComAgent Network Preference.

Available Local Server Groups: MP_SG, NO1_SG, DR_NO1_SG

Assigned Local Server Groups:

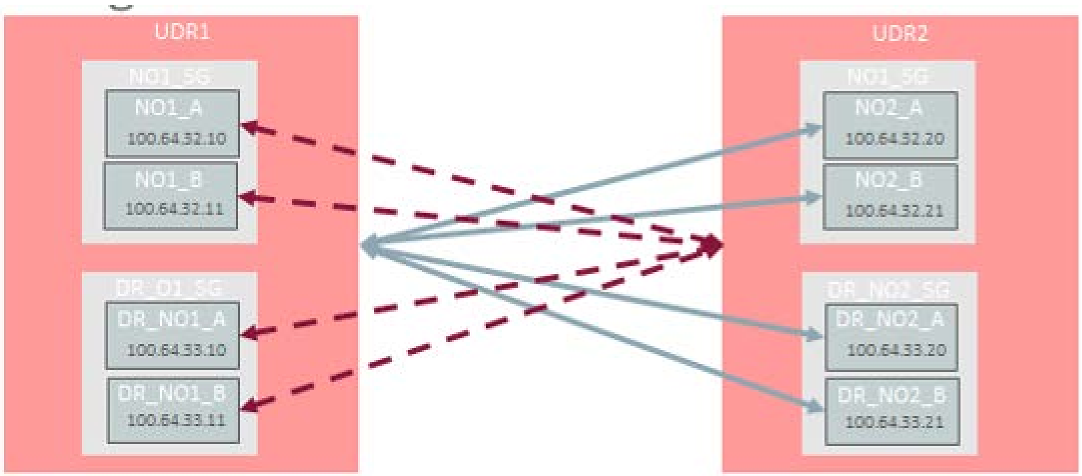
6. Configure the remaining NOs in UDR2 as HA remote server in UDR1.

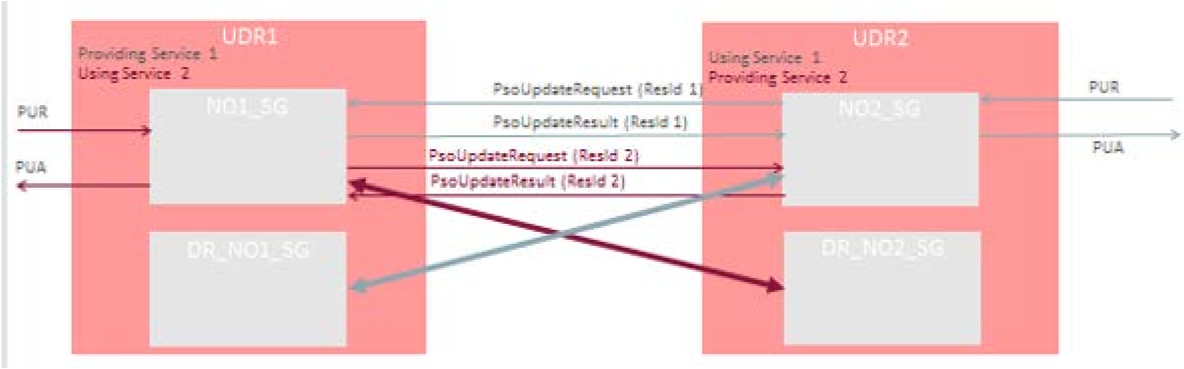


Configure the remaining NOs in UDR2 as HA remote servers in UDR1

UDR2 sends the Pool Spanning events to UDR1.

7. Repeat step 4 and step 5 for UDR1.





Now, UDR1 can send Pool Spanning events to UDR2.

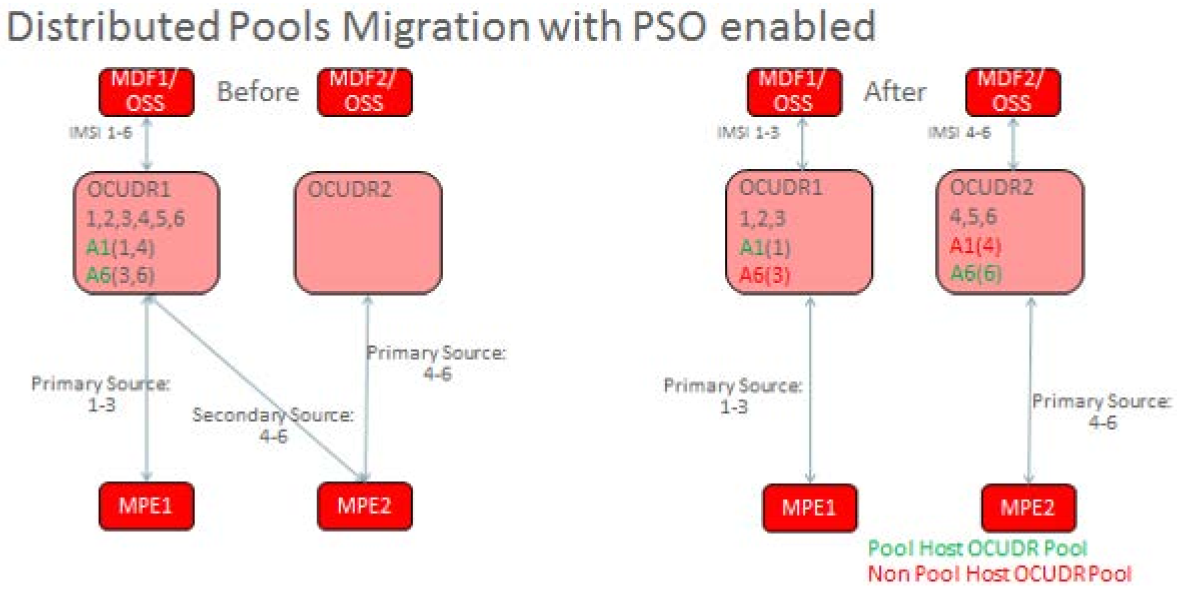
Activating a Pool Spanning Network

The following example explains the procedure to activate a pool spanning network.

Note: Once the Pool Spanning is activated, it cannot be deactivated.

In this example, UDR1 hosts all of the subscribers and pools for IMSI range 1 to 6. UDR2 is newly installed UDR which is going to host IMSI range 4 to 6 in the distributed pool network and UDR1 is going to host IMSI range 1 to 3 after the migration. UDR2 has to be configured to be the primary data source for MPE2 which at the start of the migration hosts the sessions for subscribers with IMSI range 4 to 6.

At the start of the migration, the distributed pool network is fully configured and enabled with active ComAgent connections between UDR1 and UDR2.



To activate a pool span network:

1. Upgrade UDR1 and UDR2 to the release 12.11.0.
2. Configure ComAgent on UDR1 and UDR2. See, Configuring ComAgent for UDR Machines.
3. Configure Pool Network Table on UDR1 and UDR2. See, Configuring Pool Network.
4. Add entries in the UDR Key Range table on UDR1 and UDR2 for the pool ID ranges. See, Inserting UDR for Pool Network.
5. Set **Pool Spanning Feature** to enabled.

   Signaling starts looking for Pool Spanning Pool members and Provisioning starts checking UDR Key Range table.
6. Disable provisioning for during pool migration phase.
7. Run the following command to migrate pooled subscribers in IMSI range:`udr1# o2omt –imsi 4-6 –pso`
8. Enable/Configure migration on demand on UDR2 for subscriber phase.
9. Enable provisioning on both systems.
10. Run the following command to use o2omt to migrate non-pooled subscribers in IMSI range.`udr1# o2omt –imsi 4-6`
11. Disable migration on demand.

Enabling Pool Spanning Options

You can enable the Pool Spanning feature through the Pool Spanning Options window in the main menu.

To enable the Pool Spanning feature:

1. Install and configure User Data Repository product.
2. Log in to the main menu.
3. Click **Pool Spanning**.
4. Click **Pool Spanning Options**.
5. Select the following checkbox:
   - Pool Profile Merge Enabled
   - Pool Spanning Enabled
6. Click **Apply**.

   The Pool Spanning feature is enabled now.

Table 4-2 Pool Spanning Options Fields

Element	Description
Pool Profile Merge Enabled	Enables/disables whether the Pool Profile is merged on the Non-Pool Host User Data Repository when returning the Pool Profile via Sh.
Pool Spanning Enabled	Enables/disables Pool Spanning Feature

Configuring Pool Network

You can configure the UDRs that are in the pool network from Pool Network configuration window. Each UDR is capable of being both a Pool Host UDR and a Non-Pool Host UDR at the same time. The Pool Network Configuration display window has the options to add or delete an entry.

To configure Pool Network for pool spanning:

1. In the main menu, click **Pool Spanning**.
2. Click **Pool Network Configuration**.

   The Pool Network Configuration window opens.
3. You can perform following in this window:
   - Filter UDRs based on specific criteria. See, Filtering UDR from Pool Network.
   - Insert a new UDR to configure to the pool network. See, Inserting UDR Key Range.
   - Edit an existing UDR. See, Editing UDR for Pool Network.
   - Delete a UDR. See, Deleting UDR Key Range.

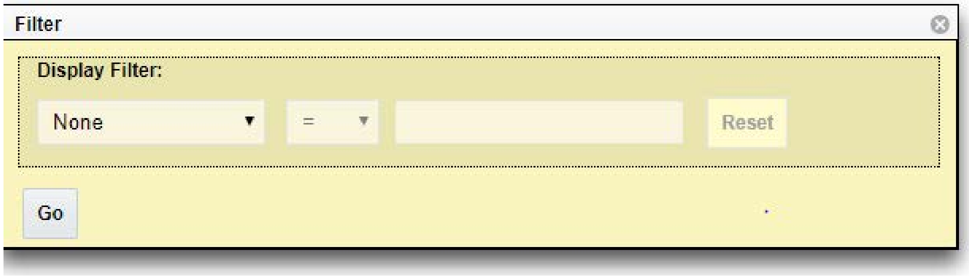
Filtering UDR from Pool Network

The filter window allows you to view or filter the UDR data. You can filter the results for specific UDR. The filter provides the results based on the UDR Name, UDR ID, and Type.

To filter the UDR from pool network:

1. Log in to the User Data Repository main menu.
2. Click **Pool Spanning**.
3. Click **Pool Network Configuration** and then click **Filter**.

   The filter window opens.



4. From the first drop-down menu, select the desired option from the following:

   - None
   - UDR Name
   - UDR ID
   - Key Type

   If you have selected **None**, the remaining fields remain disabled and no results are displayed.

5. The second field is auto-populated based on the selection in the first field.
6. (Optional) Enter any value based on the selection in the first field.
7. Click **Go** to view the results or click **Reset** to enter the filter details again.

Inserting UDR for Pool Network

You can add a new UDR to Pool Network from the Pool Network Configuration window.

To insert a new UDR to the pool network:

1. Click **Insert** on the Pool Network Configuration window.

   The Insert window opens.

2. In the **UDR Name** field, enter the UDR name from the drop-down.
3. In the **UDR ID** field, enter the UDR ID.
4. From the Type list, select the **PoolID** to define the pool ranges.
5. Click **OK** to add the UDR.

Table 4-3 Pool Network Insert Window Fields

Element	Description
UDR Name	Indicates the name assigned to represent the UDR instance with the given ID. A 15-character string. Valid characters are alphanumeric and underscore. Must contain at least one alpha and must not start with a digit.
UDR ID	A unique non-zero UDR Instance ID. The valid range is 1 to 4294967295.
Key Type	The flag indicates if UDR ID is Host or Remote. Self for Host UDR and Remote for Remote UDR. This is set to Self only for Host UDR.

Editing UDR for Pool Network

You can edit the defined UDR Key Range by using edit option in the UDR Key Range window.

To edit a UDR for Pool Network:

1. Click **Edit** on the UDR Key Range window.

   The Edit window opens.

2. In the **UDR Name** field, enter the UDR name from the drop-down menu.
3. In the **UDR ID** field, enter the UDR Instance ID.
4. From the Type list, select the **PoolID** to define the pool ranges.
5. Click **OK** to apply the changes.

You have successfully edited the UDR Key Range details.

Deleting UDR from Pool Network

You can delete the defined UDR from the pool network from Pool Network Configuration window.

To delete a UDR from pool network:

1. From the pool network configuration window, select the key range to be deleted.
2. Click **Delete**.

   A confirmation window opens.

3. Click **OK** to delete.

You have successfully deleted the UDR from pool network.

Working with UDR Key Range for Pool Spanning

The UDR Key Range allows you to configure key ranges for which User Data Repository members of the pool network that determines which UDR in the Pool Network is the Pool Host UDR for a pool.

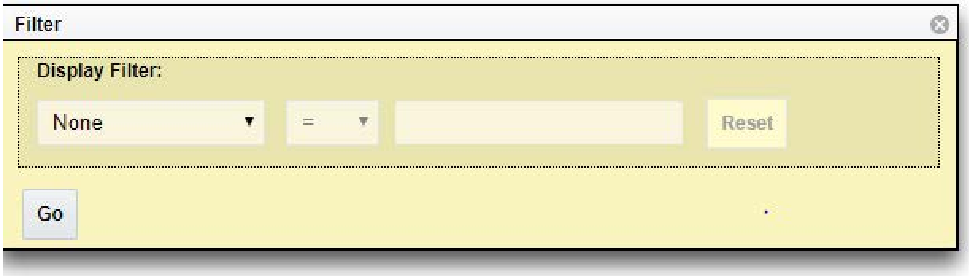
Filtering UDR Key Range

The filter window allows you to view or filter the UDR Key Range data. You can filter the results for specific Ud attributes. The filter provides the results based on the UDR Name, URD ID, and Type.

To filter the UDR Key Range data:

1. Log in to the User Data Repository main menu.
2. Click **Pool Spanning**.
3. Click **UDR Key Range** and click **Filter**.

   The filter window opens.



4. From the first drop-down, select the option from the following:

   - None
   - UDR ID
   - Key Type
   - Start Range
   - End Range

   If you have selected **None**, the remaining fields remain disabled and no results are displayed.

5. The second field is auto-populated based on the selection in the first field.
6. (Optional) Enter any value based on the selection in the first field.
7. Click **Go** to view the results or click **Reset** to enter the filter details again.

Inserting UDR Key Range

You can add a new UDR Key Range from the window.

To insert a new UDR Key Range:

1. Click Insert on the UDR Key Range window.

   The Insert window opens.

2. From the **UDR Name** field, select the UDR name from the drop-down.
3. From the Key Type list, select the **PoolID** to define the pool ranges.
4. In the **Start Range** field, enter the data range to include for the UDR Key Range.
5. In the **End Range** field, enter the data range to include for the UDR Key Range.
6. Click **OK** to add the UDR key range.

Table 4-4 UDR Key Range Elements

Element	Description
UDR Name	Indicates the name assigned to represent the UDR instance with the given ID. A 15-character string. Valid characters are alphanumeric and underscore. Must contain at least one alpha and must not start with a digit.
UDR ID	A unique non-zero UDR Instance ID. The valid range is 1 to 4294967295.
Key Type	The flag indicates if UDR Id is Host or Remote. Self for Host UDR and Remote for Remote UDR. This is set to Self only for Host UDR.
Start Range	Start of a range of data to be included in this UDR key range list. Format: Integer Range: 1-22 digits
End Range	End of a range of data to be included in this UDR key range list. Format: Integer Range: 1-22 digits

Editing UDR Key Range

You can edit the defined UDR Key Range by using edit option in the UDR Key Range window.

Warning! The UDR Key Range must be modified with caution. Any modification to the ranges would affect the already existing pool-subscriber relation. Hence, modification to UDR Key range must be followed by deletion and re-provisioning of all pools.

To edit a UDR Key Range:

1. Click **Edit** on the UDR Key Range window.

   The Edit window opens.

2. From the **UDR Name** field, select the UDR name from the drop-down.
3. From the Key Type list, select the **PoolID** to define the pool ranges.
4. In the **Start Range** field, enter the data range to include for the UDR Key Range.

5. In the **End Range** field, enter the data range to include for the UDR Key Range.
6. Click **OK** to apply the changes.

You have successfully edited the UDR Key Range.

Deleting UDR Key Range

You can delete a defined UDR Key Range from Pool Spanning network in the UDR Key Range window.

To delete a UDR Key Range:

1. From the UDR Key Range window, select the key range to be deleted.
2. Click **Delete**.

   A confirmation window opens.
3. Click **OK** to delete.

You have successfully deleted the selected UDR Key Range.

5

Simple Network Management Protocol (SNMP) Configuration

This chapter provides an overview of SNMP and describes the SNMP configuration.

Overview

Simple Network Management Protocol (SNMP) is a communication protocol that provides a method for managing TCP/IP networks, including individual network devices, and devices in aggregate. SNMP was developed by the IETF (Internet Engineering Task Force) and is applicable to any TCP/IP network, as well as other types of networks.

SNMP is an Application Program Interface (API) to the network so that general-purpose network management programs can be easily written to work with a variety of different devices. SNMP defines a client/server relationship. The client program (called the network manager) makes virtual connections to a server program (called the SNMP agent). The SNMP agent executes on a remote network device and serves information to the manager about the status of the device. The database (referred to as the SNMP Management Information Base or MIB) is a standard set of statistical and control values that are controlled by the SNMP agent.

Through the use of private MIBs, SNMP allows the extension of the standard values with values specific to a particular agent. SNMP agents can be tailored for a myriad of specific devices such as computers, network bridges, gateways, routers, modems, and printers. The definitions of MIB variables supported by a particular agent are incorporated in descriptor files that are made available to network management client programs so that they can become aware of MIB variables and their usage. The descriptor files are written in Abstract Syntax Notation (ASN.1) format.

Directives are issued by the network manager client to an SNMP agent. Directives consist of the identifiers of SNMP variables (referred to as MIB object identifiers or MIB variables), along with instructions to either get the value for the identifier or set the identifier to a new value.

The SNMP Standard

SNMP can be viewed as three distinct standards:

- **A Standard Message Format**

  SNMP is a standard communication protocol that defines a UDP message format.

- **A Standard Set of Managed Objects**

  SNMP is a standard set of values (referred to as SNMP objects) that can be queried from a device. Specifically, the standard includes values for monitoring TCP, IP, UDP, and device interfaces. Each manageable object is identified with an official name, and also with a numeric identifier expressed in dot-notation.

- **A Standard Way of Adding Objects**

  A standard method is defined to allow the standard set of managed objects to be augmented by network device vendors with new objects specific for a particular network.

SNMP Message Types

Four types of SNMP messages are defined:

- A `get` request returns the value of a named object. Specific values can be fetched to determine the performance and state of the device, without logging into the device or establishing a TCP connection with the device.
- A `get-next` request returns the next name (and value) of the next object supported by a network device is given a valid SNMP name. This request allows network managers to review all SNMP values of a device to determine all names and values that an operant device supports.
- A `set` request sets a named object to a specific value. This request provides a method of configuring and controlling network devices through SNMP to accomplish activities such as disabling interfaces, disconnecting users, and clearing registers.
- A `trap` message is generated asynchronously by network devices, which can notify a network manager of a problem apart from any polling of the device. This typically requires each device on the network to be configured to issue SNMP traps to one or more network devices that are awaiting these traps. The four message types are all encoded into messages referred to as Protocol Data Units (PDUs), which are interchanged with SNMP devices.

Standard Managed Objects

The list of values that an object supports is referred to as the SNMP Management Information Base (MIB). MIB can be used to describe any SNMP object or portion of an SNMP hierarchy.

The various SNMP values in the standard MIB are defined in RFC-1213, one of the governing specifications for SNMP. The standard MIB includes various objects to measure and monitor IP activity, TCP activity, UDP activity, IP routes, TCP connections, interfaces, and general system description. Each of these values is associated with an official name (such as sysUpTime, which is the elapsed time since the managed device was booted) and with a numeric value expressed in dot-notation (such as 1.3.6.1.2.1.1.3.0, which is the object identifier for sysUpTime).

See Supported MIBs for a description of the use of SNMP MIBs for Policy Management. E66969 Revision 01, November 2016 21 Overview.

Configuring SNMP

This section describes how to configure SNMP using the CMP system.

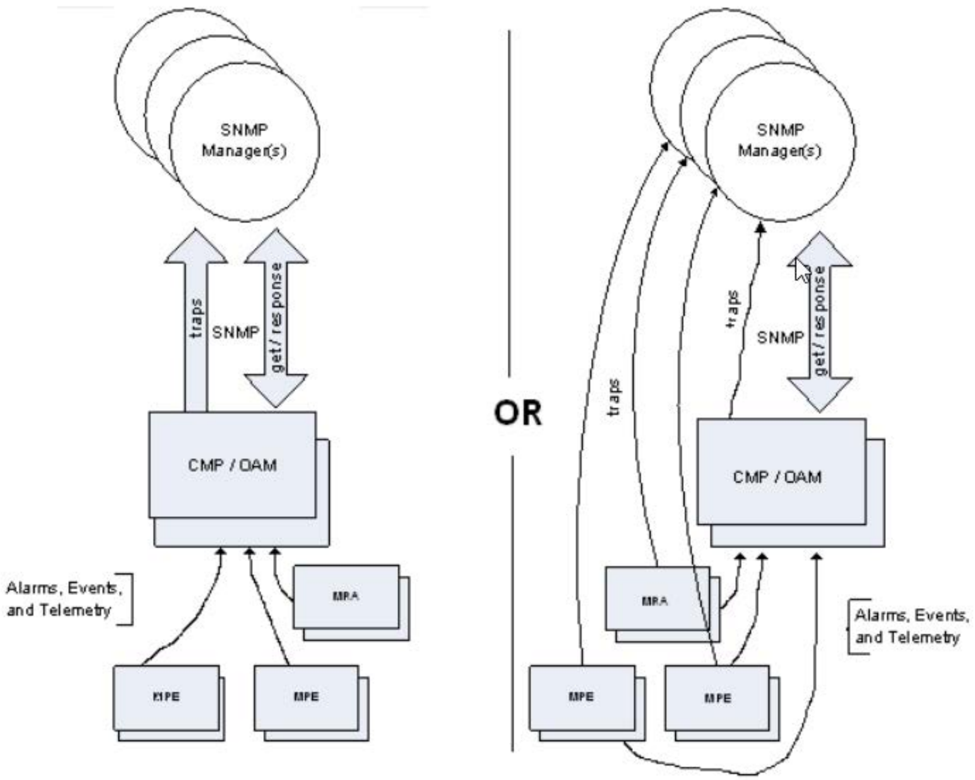
About SNMP Configuration

SNMP configuration architecture is based on using traps to notify a network management system of events and alarms that are generated by the MPE and MRA application software, and those that are generated by the underlying platforms. Alarms and telemetry data are continuously collected from the entire Policy Management network and stored in the CMP system. Alarms cause a trap to be sent as a notification of an event.

Because the underlying platform can deliver the alarms from the MPE or MRA system to the CMP system, SNMP can be configured in either of two ways:

The Policy Management system can be configured so that the CMP system is the source of all traps (the left side of Figure SNMP Configuration).

The Policy Management system can be configured to allow each server to generate its own traps and deliver them to the SNMP management servers the right side of the following figure.



The Traps from individual Servers option (see Configuring SNMP Settings) determines the mode in which the SNMP notifications operate. When enabled, each server generates traps and the Policy Management system operates as shown on the right side of Figure. SNMP configuration is pushed from the CMP system to the managed servers in the network.

SNMP Versions

Note: SNMP version 1 (SNMPv1) is not supported.

SNMP version 2c (SNMPv2c) and SNMP version 3 (SNMPv3) are supported. On the SNMP Setting Edit page:

- When you configure SNMPv2c, you must use a Community Name that is not public or private.
- When you configure SNMPv3, you must enter an Engine ID, a Username, and Password for the SNMPv3 user.

Configuring SNMP Settings

You can configure SNMP settings for the CMP system and all Policy Management servers in the topology network. You can configure the Policy Management network such that the CMP system collects and forwards all traps or such that each server generates and delivers its own traps.

Note: SNMP settings configuration must be done on the active CMP server in the primary cluster. A warning displays if the login is not on the active primary CMP system.

To configure SNMP settings:

1. Log in to the CMP system using a username with administrator privileges.
2. From the Platform Setting section of the navigation pane, select **SNMP Setting**.

   The SNMP Settings page opens.
3. Click **Modify**.

   The Edit SNMP Settings page opens.
4. For each SNMP Manager, enter a valid hostname or an IPv4IPv4/IPv6 address.

   The Hostname/IP Address field is required for an SNMP Manager to receive traps and send SNMP requests.

   The field has the following restrictions:

   - A hostname should include only alphanumeric characters.
   - Maximum length is 20 characters.
   - Case insensitive (uppercase and lowercase are treated as the same). By default, these fields are blank.

   Note: Note: The IPv6 address is not supported.
5. (Optional) You can configure a port for each SNMP Manager by entering a port value between 1 and 65535 in the Port field. If left blank, the default value is 162.

6. From the Enabled Versions list, select one of the following versions:

   - SNMPv2c
   - SNMPv3
   - SNMPv2c and SNMPv3 (default)

7. If you selected SNMPv2c or SNMPv2c and SNMPv3 from the Enabled Versions list, configure the following:

   a. **Traps EnabledTraps Enabled**

      Specifies whether sending SNMPv2 traps is enabled. The default is enabled.

      Note: To use the SNMP Trap Forwarding feature, enable this option.

   b. **Traps from individual Servers**

      Specifies whether sending SNMPv2 traps from individual servers is enabled. If disabled, SNMPv2 traps are only sent from the active CMP system only. The default is disabled.

      Note: Note: To use the SNMP Trap Forwarding feature, disable this option.

   c. **SNMPv2c Community Name**

      Enter the SNMP read-write community string.

      This field has the following restrictions:

      - The field is required if SNMPv2c is enabled.
      - The name can contain alphanumeric characters and cannot exceed 31 characters in length.
      - The name cannot be either `private` or `public`.

        The default value is `snmppublic`.

8. If you have selected **SNMPv3** or **SNMPv2c**, and **SNMPv3** from the **Enabled Versions** list, configure the following:

   a. **SNMPv3 Engine ID** — Enter an Engine ID for SNMPv3. The Engine ID can be 10 to 64 digits long and must use only hexadecimal digits (0 to 9 and a to f). The default is no value (null).

   b. **SNMPv3 Security Level** — Select the level of SNMPv3 authentication and privacy from the list:

      - **No Auth No Priv** — Authenticate using the Username. No Privacy.
      - **Auth No Priv** — Authenticate using MD5 or SHA1 protocol.
      - **Auth Priv (default)** — Authenticate using MD5 or SHA1 protocol. Encrypt using the AES or DES protocol.

   c. **SNMPv3 Authentication Type** — Select an SNMPv3 authentication protocol from the list:

  - **SHA-1** — Use Secure Hash Algorithm authentication.
  - **MD5 (default)** — Use Message Digest authentication.

d. **SNMPv3 Privacy Type** — Select an SNMPv3 privacy protocol from the list:

  - **AES (default)** — Use Advanced Encryption Standard privacy.
  - **DES** — Use Data Encryption Standard privacy.

e. **SNMPv3 Username** — Enter a username. The username can contain 0 to 32 characters and must only contain alphanumeric characters.

The default is `TekSNMPUser`.

f. **SNMPv3 Password** — Enter an authentication password. The password must contain between 8 and 64 characters and can include any character. The default is `snmpv3password`.

Note: The SNMPv3 password is also used for `msgPrivacyParameters`.

9. Select **Traps Enabled** to enable sending SNMPv2 traps. The default is enabled. Uncheck the check box to disable sending SNMPv2 traps.

Note: To use the SNMP Trap Forwarding feature, enable this option.

10. Select **Traps from individual Servers** to enable sending traps from each individual server. The default is disabled. Uncheck the checkbox to send traps from the active CMP system only.

Note: To use the SNMP Trap Forwarding feature, disable this option.

11. Enter the SNMPv2c Community Name.

This is the SNMP read-write community string. This field has the following restrictions:

- The field is required if SNMPv2c is enabled.
- The name can contain alphanumeric characters and cannot exceed 31 characters in length.
- The name cannot be either `private` or `public`.

The default value is `snmppublic`.

12. Enter the SNMPv3 Engine ID.

This is the configured Engine ID for SNMPv3. This field has the following restrictions:

- The field is required if SNMPv3 is enabled.
- The Engine ID uses only hexadecimal digits (0 to 9 and a to f).
- The length can be from 10 to 64 digits.

The default value is no value (null).

13. Select the SNMPv3 Security Level (SNMPv3 Authentication and Privacy) from the list:

    - **No Auth No Priv** — Authenticate using the Username. No Privacy.
    - **Auth No Priv** — Authenticate using MD5 or SHA1 protocol.
    - **Auth Priv** — [default] Authenticate using MD5 or SHA1 protocol. Encrypt using the AES or DES protocol.

14. Select the SNMPv3 Authentication Type (Authentication protocol for SNMPv3) from the list:

    - **SHA-1** — Use Secure Hash Algorithm authentication.
    - **MD5** — [default] Use Message Digest authentication.

15. Select the SNMPv3 Privacy Type (Privacy Protocol for SNMPv3) from the list:

    - **AES** — [default] Use Advanced Encryption Standard privacy.
    - **DES** — Use Data Encryption Standard privacy.

16. Enter the SNMPv3 Username.

    This field has the following restrictions:

    - The field is required if SNMPv3 is enabled.
    - The name must contain alphanumeric characters and cannot exceed 32 characters in length.

    The default value is `TekSNMPUser`.

17. Enter the SNMPv3 Password. This value is the Authentication password for SNMPv3 and is also used for `msgPrivacyParameters`.

    This field has the following restrictions:

    - The field is required if SNMPv3 is enabled.
    - The length of the password must be between 8 and 64 characters and can include any character. The default value is `snmpv3password`.

18. Click **Save**.

The SNMP settings for the network are configured.

Configuring Different EngineID on Different Servers

You can configure EngineIDs on different servers such as NOAMA, NOAMB, SOAMA, SOAMB, MP1, MP2, and MP3.

To configure an EngineID on a Server:

1. On Active NOAMP interface, in the main menu, go to **Administration** > **Remote Servers** > **SNMP Trapping** and select **Traps from Individual Servers** to enable the configuration..

2. Run the following command for disabling the replication of the SNMP configuration table across different NOAM, SOAM, and MP servers so that the change in one server does not replicate across different servers.

   For example, run the following command in Active NOAM to disable the replication of SNMP configuration: `iset -fexcludeTables=SnmpCfg NodeInfo where 1=1`

3. Change the EngineID on all the servers by disabling them.

   Note: You need not disable for Active NOAM server.

   In case if there is a disaster recovery site then perform the same steps in each server (i.e. NOAMA, NOAMB, SOAMA, SOAMB, MP1, MP2, MP3, MP4 and so on in Disaster recovery site).

   Run the following commands to disable the servers:

   - Active NOAM:`iset -fengineId="123456NOAMA" SnmpCfg where 1=1;`
   - Standby NOAM:`prod.dbdown -iprod.dbupiset -fengineId="123456NOAMB" SnmpCfg where 1=1;prod.dbdown -iprod.start`
   - Standby SOAM B:`prod.dbdown -iprod.dbupiset -fengineId="123456SOAMB" SnmpCfg where 1=1;prod.dbdown -iprod.start`
   - Active SOAM A:`prod.dbdown -iprod.dbupiset -fengineId="123456SOAMA" SnmpCfg where 1=1;prod.dbdown -iprod.start`
   - MP1:`prod.dbdown -iprod.dbupiset -fengineId="12345678MP1" SnmpCfg where 1=1;prod.dbdown -iprod.start`
   - MP2:`prod.dbdown -iprod.dbupiset -fengineId="12345678MP2" SnmpCfg where 1=1;prod.dbdown -iprod.start`

4. Verify the entry for the EngineID on all the servers.

   For example, run the following command on the `/etc/snmp/snmpd.conf` file.`cat /etc/snmp/snmpd.conf | grep -i engineID`

Getting Object Identifier (OID) for Different Objects using MIB File

This section explains the details of the MIB used in UDR and how to generate an OID using SNMP command using an example.

Run the following command to retrieve the OIDs for dfferent objects using MIB file:

```
snmptranslate -Tz -m /usr/TKLC/plat/etc/snmp/mib/
tklc_tpdAlarms.mib
```

All UDR related MIB files are stored in following paths in NOAM, SOAM, or MP servers:

- /usr/TKLC/udr/mibs

- /usr/TKLC/plat/etc/snmp/mib

Example for generating Object ID by `cmsnmpsa comcol` Process.

When alarm is generated, the entries are put in to the AppEventLog tables. The merged traps entries from the AppEventLog_001 are sent to the snmpagents. The OID is calculated as follows:

```
entry in the AppEventLog for the Alarm raised
for eventNumber 13071:mysql> select * from AppEventLog_001 where
eventNumber = 13071;| part | srcNode | severity | timeStamp |
task | eventNumber | instance | eventData |errInfo |
additionalInfo ||0| A1173.045 | ^^ | 1611745169468401377|
udrprov|13071 | PROV| |IDB_ENDTBL/CLR cursor at end of table
[IdbBaseIter.cxx.cmf:209]^^ [26904:ProvController.C:
```

```
OBJECT
OID:<ORIGINAL_OID>.<TRAPFLAG>.<SRCNODEID>.<ALARMNUMBER>.<SIZEOFI
NSTANCE>.<INSTANCEDATA>
```

For example, following OID for alarm 13071 is shown below:

```
Final OID for
Object, eagleXgUdrAlarmInstance in Trap for eventNumber
13071:1.3.6.1.4.1.323.5.3.32.1.1.3.5.1.4.1.164.149.45.0.13071.4.
80.82.79.86
```

Following is the OID from the MIB file, which got by using the `snmptranslate` command:

```
ORIGINAL_OID of eagleXgUdrAlarmInstance =
1.3.6.1.4.1.323.5.3.32.1.1.3.5.1.4TRAPFLAG = 1, either 0 or 1,
```

If the Local trap is disabled, it is set to 1 in the trap OID,

```
SRCNODEID =
164.149.45.0,
```

The value is calculated based on the entry present in the srcNode which is A1173.045

If nodeId is A1173.045, then:

```
Level = AClusterNumber = 1173MemberNumber =
045
```

RUNID can be retrieved from the server using the following command in NOAM environment: `echo $RUNID`

For reference, see the following example where RUNID equals 0:

```
(level << 4)|
((ClusterNumber & 0xf00) >> 8) = ('A' << 4)|((1173 & 0xf00) >>
8) = (0x41 << 4)|((0x495 & 0xf00) >> 8) = 0xA4 = 1641173 and
0xff = 0x495 and 0xff = 14945 and 0xff = 0x2D and 0xff 0x2D =
45RUNID and 0xff = 0 & 0xff = 0
```

Here, RUNID is the environment variable retrieved from the NOAM, SOAM, and MP servers using the command, `echo $RUNID`.

0xff is an integer value used for masking purpose to retrieve only the first byte of the RUNID. Since the value of the RUNID is 0, the final result of the & operation is 0.

Finally, 164.149.45.0 is used as SRCNODEID.

If the Local trap is enabled then SRCNODEID is set to 0.0.0.0`ALARMNUMBER = 13071`

This is the eventNumber/alarmid from the AppEventLog tables.

```
SIZEOFINSTANCE =
4
```

This is the size of the instance string from the AppEventLog_001 table for the alarm`INSTANCEDATA = 80.82.79.86 (PROV)`

This is the value of the instance which is used from the AppEventLog_001 table.

Here, the value of the instance is PROV decimal value which is appended to the final trap OID.

Therefore, P = 80; R = 82; O = 79; V = 86

Finally, 80.82.79.86 is used as INSTANCEDATA

SNMP Trapping

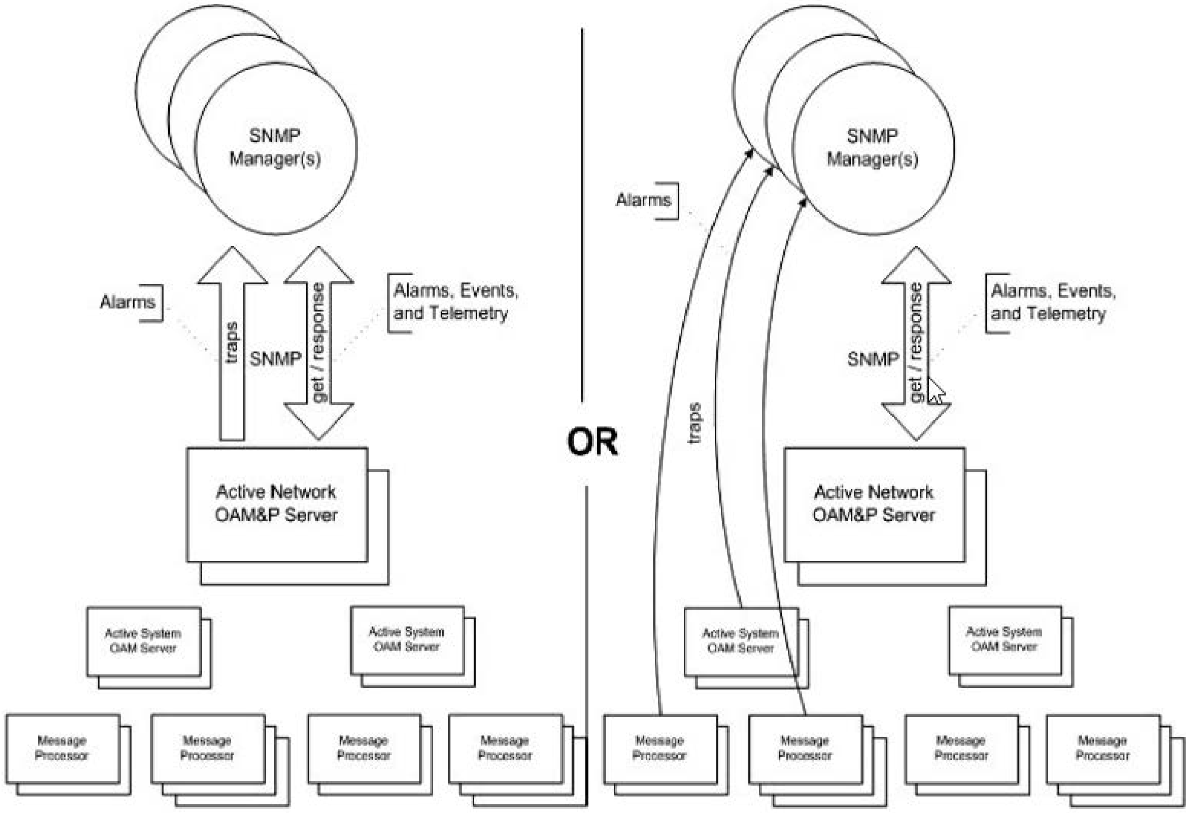
The SNMP Trapping page enables the user to configure up to five remote managers to receive traps using the industry-standard Simple Network Management Protocol (SNMP). The user can choose versions v2c, v3, or both along with the typical security parameters associated with each of the versions. In addition, traps from individual servers can be enabled from this view.

Note: The SNMP Manager is provided by the customer.

The SNMP agent is responsible for SNMP-managed objects. Each managed object represents a data variable. A collection of managed objects is called a Management Information Base (MIB). In other words, a MIB is a database of network management information that is used and maintained by the SNMP protocol. The MIB objects contain the SNMP traps that are used for alarms; a readable SNMP table of current alarms in the system; and a readable SNMP table of KPI data.

By default, system-wide traps are sent from the active Network OAM&P server while site-specific traps are sent from active Site OAM servers. Alternately, functionality may be enabled that allows individual servers to send traps, in which case individual servers interface directly with SNMP managers.

Note: Note: Only the Active Network server allows SNMP administration.



The application sends SNMP traps to SNMP Managers that are registered to receive traps. IP addresses and authorization information can be viewed and changed using the SNMP administration page. For SNMP to be enabled, at least one Manager must be set up.

SNMP Administration Elements

On the active network OAM&P server, the SNMP Administration page provides for the configuration of SNMP services.Table 7 describes the elements of the SNMP Administration page.

Table 5-1   SNMP Administration Elements

Element	Description	Data Input Notes

Table 5-1 (Cont.) SNMP Administration Elements

Manager 1	Manager to receive SNMP traps and Valid IP address or a valid hostname send requests. It could be a valid IP address or a valid hostname.	Valid IP address or a valid hostname IPv4 addresses are 32 bits, represented in a dot-decimal notation like this: x.x.x.x where each x (called an octet) is a decimal value from 0 to 255. They are separated by periods. For example, 1.2.3.4 and 192.168.1.100 are valid IPv4 addresses. IPv6 addresses are 128 bits, represented in a colon-hexadecimal notation like this: z:z:z:z:z:z:z:z where each z is a group of hexadecimal digits ranging from 0 to ffff. They are separated by colons. Leading zeros may be omitted in each group. "::" can be used (at most once) in an IPv6 address to represent a range of as many zero fields as needed to populate the address to eight fields. So the IPv6 address 2001:db8:c18:1:260:3eff:fe47:1530 can also be represented as 2001:0db8:0c18:0001:0260:3eff:fe47:1530 and the IPv6 address ::1 is the same as 0000:0000:0000:0000:0000:0000:0000:0001 Hostname Format: Alphanumeric [a to z, A to Z, 0 to 9] and minus sign (-) Hostname Range: 1 to 255-character string
Manager 2	Manager to receive SNMP traps and send requests. It could be a valid IP address or a valid hostname.	See description for Manager 1.
Manager 3	Manager to receive SNMP traps and send requests. It could be a valid IP address or a valid hostname.	See description for Manager 1.
Manager 4	Manager to receive SNMP traps and send requests. It could be a valid IP address or a valid hostname	See description for Manager 1.

Table 5-1 (Cont.) SNMP Administration Elements

Manager 5	Manager to receive SNMP traps and send requests. It could be a valid IP address or a valid hostname.	See description for Manager 1.
Enabled Versions	Enables the specified version(s) of Format: Pulldown list SNMP. Options are: • SNMPv2c: Allows SNMP service SNMPv3 only to managers with SNMPv2c authentication. • SNMPv3: Allows SNMP service only to managers with SNMPv3 authentication. • SNMPv2c and SNMPv3: Allows SNMP service to managers with either SNMPv2c or SNMPv3 authentication. This is the default.	Format: Pulldown list SNMP. Range: SNMPv2c, SNMPv3, or SNMPv2c and SNMPv3 Default: SNMPv2c and SNMPv3
Traps Enabled	Enables or disables SNMP trap. The GUI user may selectively disable sending autonomous traps to SNMP managers when alarms are raised. The default is enabled. Access to alarm and KPI tables is not affected by this setting.	Format: Checkbox output. Range: Enabled or Disabled Default: Enabled
Traps from Individual Servers	Enables or disables SNMP traps from individual servers. If enabled, the traps are sent from individual servers, otherwise, traps are sent from the Network OAM&P server.	Format: Checkbox Range: Enabled or Disabled Default: Disabled
SNMPV2c Read-Only Community Name	Configured Read-Only Community Name (SNMPv2c only). Public is the default. This field is required when Name SNMPv2c is enabled in Enabled Versions. The length of community name should be less than 32 characters.	Format: Alphanumeric [a to z, A to Z, 0 to 9] Range: 1 to 31 characters Default: snmppublic **Note:** The Community Name cannot equal Public or Private.
SNMPV2c Read-Write Community Name	Configured Read-Write Community Name (SNMPv2c only). Public is the default. This field is required when SNMPv2c is enabled in Enabled Versions. The length of community name should be less than 32 characters.	Format: Alphanumeric [a to z, A to Z, 0 to 9] Range: 1 to 31 characters Default: snmppublic **Note:** The Community Name cannot equal Public or Private.

Table 5-1 (Cont.) SNMP Administration Elements

SNMPv3 Engine ID	Configured Engine ID (SNMPv3 only). This field is required SNMPv3 is enabled in Enabled Versions. A unique Engine ID is generated by default.	Format: Hex digits 0 to 9 and a to f Range: 10 to 64 characters Default: A unique Engine ID value
SNMPv3 Username	Specifies an authentication username (SNMPv3 only). The default is TekSNMPUser. This field is required when SNMPv3 is enabled in Enabled Versions.	Format: Alphanumeric [a to z, A to Z, 0 to 9] Range: 1 to 32 characters Default: TekSNMPUser
SNMPv3 Security Level	Sets authentication and privacy options (used for SNMPv3 only).	Format: Pulldown menu Range: • **No Auth No Priv:** Authenticate using the username. No Privacy. • **Auth No Priv:** Authenticate using the MD5 or SHA1 protocol. No Privacy. • **Auth Priv:** Authenticate using the MD5 or SHA1 protocol. Encrypt using the AES or DES protocol. This is the default value. Default: Auth Priv
SNMPv3 Authentication Type	Sets authentication protocol (used for SNMPv3 only).	Format: Pulldown list Range: SHA-1 or MD5 Default: SHA-1
SNMPv3 Privacy Type	Sets privacy protocol (used for SNMPv3 only). This field is required SNMPv3 Privacy Type when SNMPv3 Security Level is set to Auth Priv.	Format: Pulldown menu Range: • **AES:** Use Advanced Encryption Standard privacy. • **DES:** Use Data Encryption Standard privacy. Default: AES
SNMPv3 Password	Authentication password set up for the user specified in SNMPv3 Username (used for SNMPv3 only). This field is required when SNMPv3 is enabled and privacy is enabled at SNMPv3 Security Level.	Format: Any characters Range: 8 to 64 characters

Adding an SNMP manager

Use this procedure to add an SNMP Manager:

1. Select **Administration** > **Remote Servers** > **SNMP Trapping**.

   The SNMP Trapping page opens.

2. Update Enabled Versions as appropriate.

   For more information about Enabled Versions, or any field on this page, see SNMP administration elements.

3. Select an empty Manager field, and populate it with the hostname or IP address of the SNMP manager.
4. (Optional) Enable traps from individual servers.
5. (Optional) For SNMPv2c managers, change the SNMPV2c Read-Only Community Name.
6. (Optional) For SNMPv2c managers, change the SNMPV2c Read-Write Community Name.
7. (Optional) For SNMPv3 managers, select an SNMPv3 Security Level, and change:
   - SNMPv3 Engine ID
   - SNMPv3 Authentication Type
   - SNMPv3 Privacy Type
8. For SNMPv3 managers with user authentication enabled, configure SNMPv3 Username.
9. For SNMPv3 managers with privacy enabled, configure SNMPv3 Password.
10. Click **OK** or **Apply** to submit the information.

The new manager and related settings are saved and activated.

Viewing SNMP trap settings

Use this procedure to view SNMP trap settings:

Select **Administration** > **Remote Servers** > **SNMP Trapping**

The SNMP Trapping page opens.

The page lists all SNMP options on the system.

Updating SNMP trap settings

Use this procedure to update SNMP trap settings:

1. Select **Administration** > **Remote Servers** > **SNMP Trapping**.

   The SNMP Trapping page opens.

2. Update SNMP trap settings as needed.

   For more information, see SNMP administration elements.

3. Click **OK** or **Apply** to submit the information.

The SNMP trap changes are saved, and activated.

Deleting SNMP trap managers

Use this procedure to remove one or more SNMP trap managers:

1. Select **Administration** > **Remote Servers** > **SNMP Trapping**.

   The SNMP Trapping page opens.

2. Delete the SNMP hostname and IP addresses from the Manager fields for which you want traps removed.

3. Click **OK** or **Apply**.

The SNMP configuration changes are saved. If the SNMP manager hostnames and IP addresses are cleared from all Manager fields, the SNMP feature is effectively disabled.

6

Ud Client Feature Configuration

This chapter describes the procedures to configure Ud Client feature for User Data Repository (UDR).

Overview

The Ud Client allows the UDR to function as a User Data Convergence Front-End element, which allows UDR to access, retrieve, and update subscriber records in a third-party UDR network other than UDR. The subscriber profile retrieved from the third-party UDR network other than Oracle Communications User Data Repository is stored in a format that is transparent to Policy Management (PM), thus providing transparent support for all existing quota management capabilities.

UDR interfaces with an embedded third-party subscriber database and provides support for all quota management use cases. UDR also preserves all existing quota management capabilities.

Ud client facilitates UDR to access subscriber records in a separate off-board subscriber database. Whenever a subscriber record is not available in the UDR, then UDR uses this interface to retrieve the record from an off-board subscriber database. After retrieving, the record is cached locally in the UDR until it is for the application. UDR also subscribe for notifications from the separate subscriber database to receive notifications in the event that the subscriber record is modified in the other database.

UDR leverages LDAP to retrieve and update subscriber profiles. For more information, refer to the 3GPP TS 29.335, User Data Repository Access Protocol over the Ud Interface, Release 12.11.0. This occurs when PM performs the first attempt to access a particular profile for a subscriber via Sh. Upon retrieving the subscriber profile via LDAP search, UDR converts the profile data from the format provided by the off-board database to the internal format that is cached in the UDR. This involves mapping database fields to the subscriber profile, based on configuration data in the SEC. The subscriber data is cached in UDR and made available to any other application that chooses to access the data from UDR that includes PM, and other third-party applications that are integrated to UDR.

You can refresh the profile for a subscriber based on a configured refresh interval. When enabled, the profile for a subscriber is refreshed if the specified refresh interval duration has elapsed since the last refresh time. This refresh is triggered by the first Sh request associated with the subscriber after the refresh interval has elapsed.

Each time the subscriber profile is refreshed, the subscriber profile in the UDR is updated to align with the current information provided by the off-board database. The UDR generates a Push-Notification-Request (PNR) in scenarios where a change has occurred in the subscriber profile and the subscriber has an active subscription for notifications. A PNR is not be generated if the refresh does not result in changes to the subscriber profile.

UDR leverages SOAP to subscribe for notifications for any updates that occur to the specified profile for the subscriber in the off-board database. For more information, refer to the 3GPP TS 29.335, User Data Repository Access Protocol over the Ud Interface, Release 12.11.0. Whenever UDR retrieves a profile for the subscriber from the off-board database, it subscribes to notifications for any updates that occur for that subscriber. UDR supports a configurable expiry limit for SOAP subscriptions, which is negotiated with the off-board subscriber database in response to subscription. UDR monitors the expiration of active subscriptions, and initiates a re-subscription before the current subscription expires.

Profiles remain in the UDR database until one of the following events occur. When any of these events occur, the UDR notifies the PM via Sh PNR if there is an active subscription for notifications associated with the subscriber:

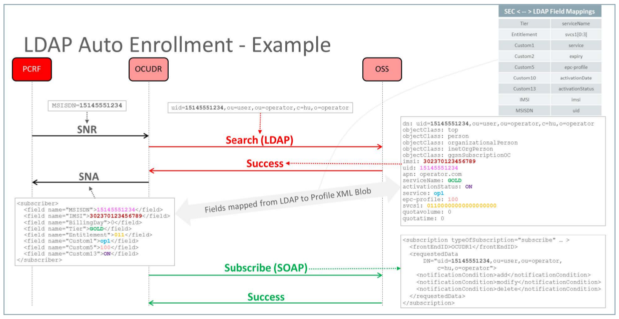
- A periodic LDAP profile refresh is performed, when the off-board database indicates that the subscriber does not exist.
- A SOAP subscription for notifications is performed, when the off-board database indicates that the subscriber does not exist.
- The UDR provisioning or bulk import interfaces are used to delete a subscriber.

Subscriber Data Schema

You can query for subscriber profile in the Ud client interface provided by UDR. This feature expects that the structure of the subscriber data in the off-board database aligns with the hierarchal structure outlined by 3GPP TS 32.181, User Data Convergence (UDC) Framework for Model Handling and Management, Release 12 and 3GPP TS 132.182, User Data Convergence (UDC) Common baseline Information Model, Release 9. Whenever UDR retrieves a subscriber record from the off-board database, it internally translates it into the internal XML format that it uses to store subscriber data in customer production networks.

UDR applies the internal structure which involves having a base profile and extended with entity data used to categorize data for different applications and functions. This structure emulates a hierarchy as the subscriber data is mapped to an LDAP interface. The base level of the hierarchy is associated with the subscriber profile. A single level request maps to the set of requested entities that are associated with the subscriber profile. Whenever the Ud client retrieves a subscriber profile from the off-board database, it leverages this configuration data to determine how the data schema provided by the off-board database maps to the UDR subscriber profile.

An example of a Subscriber Search request is outlined in the following figure. It illustrates how the GUI configuration facilitates mapping between the various fields from the LDAP response to fields within the subscriber profile.


LDAP Auto Enrollment - Example
PCRF
OCUDR
OSS
MSISDN=15145551234
uid=15145551234,ou=user,ou=operator,c=hu,o=operator
SNR
Search (LDAP)
Success
SNA
Fields mapped from LDAP to Profile XML Blob
Subscribe (SOAP)
Success

SEC < -- > LDAP Field Mappings	
Tier	serviceName
Entitlement	svcs1[0:3]
Custom1	service
Custom2	expiry
Custom5	epc-profile
Custom10	activationDate
Custom13	activationStatus
IMSI	imsi
MSISDN	uid

```
dn: uid=15145551234,ou=user,ou=operator,c=hu,o=operator
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: ggsnSubscriptionOC
imsi: 302370123456789
uid: 15145551234
apn: operator.com
serviceName: GOLD
activationStatus: ON
service: op1
epc-profile: 100
svcs1: 01100000000000000000
quotavolume: 0
quotatime: 0
```

```
<subscriber>
  <field name="MSISDN">15145551234</field>
  <field name="IMSI">302370123456789</field>
  <field name="BillingDay">0</field>
  <field name="Tier">GOLD</field>
  <field name="Entitlement">011</field>
  <field name="Custom1">op1</field>
  <field name="Custom5">100</field>
  <field name="Custom13">ON</field>
</subscriber>
```

```
<subscription typeOfSubscription="subscribe" … >
  <frontEndID>OCUDR1</frontEndID>
  <requestedData
      DN="uid=15145551234,ou=user,ou=operator,
          c=hu,o=operator">
    <notificationCondition>add</notificationCondition>
    <notificationCondition>modify</notificationCondition>
    <notificationCondition>delete</notificationCondition>
  </requestedData>
</subscription>
```

In this example, the following mappings are configured:

- UDR Tier maps to LDAP Field serviceName
- UDR Entitlement maps to LDAP Field svcs1[0:3] {that is, 3 char from svcs is udr Entitlement}
- UDR IMSI maps to LDAP Field imsi
- UDR MSISDN maps to LDAP uid
- UDR Custom1 maps to LDAP Field service
- UDR Custom2 maps to LDAP Field expiry
- UDR Custom5 maps to LDAP Field epc-profile
- UDR Custom10 maps to LDAP Field activationDate
- UDR Custom13 maps to LDAP Field activationStatus

LDAP Connection Establishment, Authentication, and Requests

UDR is responsible for establishing the LDAP connection from its Ud client to the off-board database. This is performed as outlined in section 5.2 of 3GPP TS 29.335, User Data Repository Access Protocol over the Ud Interface, Release 12, with UDR functioning in the role of the FE. A TCP connection is initiated, which can be secured by leveraging IPSec. UDR supports multiple simultaneous connections, in order to increase overall throughput.

The LDAP session is initiated with an LDAP BindRequest message, as outlined in IETF RFC4513, LDAP Authentication Methods and Security Mechanisms, June 2006 and 3GPP TS 29.335, User Data Repository Access Protocol over the Ud Interface, Release 12.11.0. Either the unauthenticated authentication mechanism of a simple bind or the name/password authentication mechanism of a simple bind is supported to authenticate the request, as specified in IETF RFC4513, LDAP Authentication Methods and Security Mechanisms, June 2006. UDR provides a configuration interface that allows username/password credentials to be managed and stored, which is used when it initiates an LDAP connection with the off-board database.

Once the LDAP connection has been authenticated, UDR can generate a request for subscriber data, as outlined in section 6 of 3GPP TS 29.335, User Data Repository Access Protocol over the Ud Interface, Release 12.11.0. This includes the ability to query subscriber records using LDAP messages. UDR translates the subscriber data that is received from the off-board database to its internal XML format, as described in section 4.1.1. The ability to create, update, and delete subscriber data in the off-board database is outside the scope of this feature.

When it comes time to remove the connection, then an LDAP UnbindRequest message is processed (assuming a BindRequest was used to initiate the connection), as outlined in section 5.3 of 3GPP TS 29.335, User Data Repository Access Protocol over the Ud Interface, Release 12.11.0.

7

Enabling and Configuring Ud Client

Ud Client integrates into the mainline User Data Repository product release stream and is available with UDR product by default. There are no specific dependencies on Policy Management behavior, so this feature is compatible with any Policy Management product release that is supported in combination with the release that contains this feature.

To enable and configure Ud Client feature:

1. Install and configure User Data Repository.

   See the UDR Installation document at: https://docs.oracle.com/en/industries/communications/user-data-repository/index.html

2. To activate the Ud Client feature, select **Ud Client Options** from the main menu, and then select the **Ud Client Enabled** checkbox.

3. Configure **Ud Client Options**. See Enabling Ud Client Options.

4. Configure LDAP/SOAP connection details to the Ud Server. See Configuring Ud Remote Server.

5. Configure the LDAP attribute to UDR subscriber Profile key field mappings. See Modifying Ud Client Key Details.

6. Configure the LDAP attribute to UDR subscriber Profile non-key field mappings. See Configuring Ud Client Attribute Map SEC.

7. Check the Ud Client Connection Status to ensure that LDAP and SOAP (if SOAP is configured) connections are enabled and connected.

8. If the Admin State for the LDAP and/or SOAP connections is disabled, then select the header row of the connection type and click Enable.

The Ud Client is now enabled and configured. If connections do not go into the InService status, check alarms. Ud Client measurements and the Event History for further information if the connections are not established.

Enabling Ud Client Options

The Ud Client Options window allows you to enable the Ud Client feature. The Ud Client Options display window is used to configure values for the Ud Client options. Ud Client Options controls the functionality of the Ud Client. You can specify values for various global parameters as per which the Ud Client functions.

The following table describes the fields in the UD Client Options window:

Table 7-1 Ud Client Option Fields

Field	Description
Ud Client Enabled	Enables or disables the Ud Client feature.
Ud SOAP Interface Enabled	Enables or disables the Ud SOAP interface. By default, this is enabled.
Send Ud SOAP Subscribe Request	Identifies whether the SOAP
Network LAN Timeout	Indicates the maximum time in milliseconds for which the Ud Client waits for a response from the Ud Server before timing out a SOAP or LDAP request when a connection is made over a LAN. The default value is 200. You can enter a value ranging between 10 and 30000 milliseconds.
Network WAN Timeout	Indicates the maximum time in milliseconds for which the Ud Client waits for a response from the Ud Server before timing out a SOAP or LDAP request when a connection is made over a WAN. The default value is 400. You can enter a value ranging between 10 and 30000 milliseconds.
SOAP Subscribe Request Expiry Time Period	Specifies the duration in seconds to set expiryTime in SOAP Subscribe request. A value of 0 indicates that no expiry time set. The default value is 0. You can enter a range between 0 and 1000000000 seconds.
SOAP Subscribe Re-subscribe Period	Specifies the duration in seconds upon which a SOAP Subscribe request is periodically re-sent to renew the subscription for a subscriber. A value of 0 indicates that no renewal occurs. The default value is 0. You can enter a value ranging from 0 and1000000000 seconds.
LDAP Search Re-read Period	Specifies the duration in seconds upon which an LDAP Search request is periodically re-sent to re-read the data for a subscriber. A value of 0 indicates no renewal occurs. The default value is 0. You can enter a value ranging between 0 and 1000000000 seconds.

Table 7-1 (Cont.) Ud Client Option Fields

LDAP Retry Period No Connection	Specifies the duration in seconds upon which no LDAP connections can be established to the Ud Server, the Ud Client waits before attempting to connect again. The default value is 5. You can enter a value ranging from 0 and 1000000000 seconds.
LDAP Retry Period Link Busy	Specifies the duration in seconds upon which a busy error is returned to an LDAP request, the Ud Client waits before sending another request for the connection. The default value is 10. You can enter a value ranging between 0-1000000000 seconds
SOAP Retry Period No Connection	Specifies the duration in seconds upon which when no SOAP connections can be established to the Ud Server, the Ud Client waits before attempting to connect again. The default value is 5. You can enter a range between 0 and 1000000000 seconds.
Apply	Click **Apply** to save the changes.
Cancel	Click **Cancel** to discard the changes.

Configuring Ud Remote Server

You can configure the connection details to the Ud Server that User Data Repository uses using the Ud Remote Server Configuration window. The window allows up to three different connection end-point LDAP/SOAP pairs each for the remote Ud Server.

Note: When the Ud Client feature is enabled, only the Primary Connection must be configured.

Important: If the remote Ud Server does not support SOAP, and User Data Repository is configured to not enable the SOAP Interface, then you need not configure SOAP connections.

Table 9 describes the fields in the UD Remote Server window.

Field	Description
Ud Remote Server Name	Name of the Ud Remote Server. By default, the value is n/a. The valid range is a 64-character string.

Primary Connection	Primary connection to LDAP and SOAP Server. Host is IPv4 address. By default, the field is n/a. Port is an integer. By default it is 389, for SOAP it is 8080 URI Path for SOAP server. Path is string and is optional. Connection Type to Ud Server is over LAN or WAN. The default is WAN. Select the required value from the list.
Secondary Connection	Secondary connection to LDAP and SOAP Server.
Tertiary Connection	Tertiary connection to LDAP and SOAP Server.
LDAP Authentication Type	LDAP Authentication DN to be used. Enabled only if you have selected LDAP Authentication Type as Unauthenticated or Authenticated. By default, the value is n/a. The valid range is a 512 character string.
LDAP Authentication Password	LDAP Authentication Password used in a bind. Enabled only if you have selected LDAP Authentication Type as Authenticated. By default, the value is n/a. The valid range is a 64 character string.
SOAP Front End ID	Value of frontEndID for the UDR sent in SOAP Subscribe request By default, the value is n/a. The valid range is a 64 character string.
SOAP Service Name	(Optional) Indicates the value of serviceName for the UDR sent in SOAP Subscribe request. By default, the value is n/a. The valid range is a 64-character string.
Number of Connections	Indicates the number of connections to create to LDAP and SOAP servers. By default, eight servers are connected. The maximum servers that can be connected are 100.
Apply	Click **Apply** to save the changes.
Cancel	Click **Cancel** to discard the changes

Modifying Ud Client Key Details

The Ud Client Key Details windows is added to configure the LDAP search key mapping details and their mapping to internal subscriber Profile fields that UDR uses.

Important: You must perform a valid Ud Remote Server Configuration before defining the Ud Client Key Details, else, you receive an error and the data is not saved.

You can define multiple keys, the key used depends on the key used by the Sh request which triggered the Ud-Creation of the subscriber.

A key definition includes a Base DN, with the specific key being included in either the Ud Attribute or the filter.

> **Note:** The order in which the keys are defined is relevant. If User Data Repository has a choice of keys to use for a subscriber, the first matching key found in the order they are defined is used.

For example, when re-subscribing for a subscriber. There is no triggering Sh request to initiate the re-subscribe request, a check is performed periodically.

A subscriber is read, and the keys defined in the subscriber profile are checked with the configured key details, and the first matching defined key is used to initiate the re-subscribe request.

Following table describes the fields in the Ud Client Key Details window:

Table 7-2 Ud Client Key Details

Field	Description
Profile Field Name	Maps UDR key types of keys used to access the subscriber record in the LDAP database. At least one key pattern must be configured. Profile Field Name can be IMSI, MSISDN or NAI. Default value is n/a. Select the required value from the list.
Ud Attribute Name	Ud Attribute Name indicates attribute name to set in Search DN, and also LDAP attribute to extract value in returned response to set subscriber Profile key field. Default value is n/a. The valid range is a 64 character string.
Base DN	Base DN to be used for this key. By default, the value is n/a. The valid range is a 512 character string.

Table 7-2 (Cont.) Ud Client Key Details

Search Scope	Search Scope used for LDAP Search. Available values • Base Object • One Level • Subtree The default value is Base Object. Select the required value from the list.
Filter	The filter is LDAP Search filter sent with the request, indicating key part parameter. By default, the value is n/a. The valid range is a 256 character string.
Transform Pattern	Transform Pattern indicates the part of the key pattern to be matched. By default, the value is n/a. The valid range is a 64 character string.
Replace Pattern	Replace Pattern indicates what to replace the part indicated in Transform Pattern with. By default, the value is n/a. The valid range is a 64 character string.
Apply	Click **Apply** to save the changes.
Cancel	Click **Cancel** to discard the changes

Configuring Ud Client Attribute Map SEC

The Ud Client Key Details allows you to configure the LDAP search key mapping details and their mapping to internal subscriber Profile fields that Oracle Communications User Data Repository uses.

Important: You must perform a valid Ud Remote Server Configuration before defining the Ud Client Key Details, else, you receive an error and the data is not saved.

You can define multiple keys, the key used depends on the key used by the Sh request which triggered the Ud-Creation of the subscriber.

A key definition includes a Base DN, with the specific key being included in either the Ud Attribute or the filter.

Note: The order the keys are defined is relevant. If User Data Repository has a choice of keys to use for a subscriber, the first matching key found in the order they were defined is used.

For example, when re-subscribing for a subscriber. There is no triggering Sh request to initiate the re-subscribe request, a check is performed periodically.

A subscriber is read, and the keys defined in the subscriber profile are checked with the configured key details, and the first matching defined key is used to initiate the re-subscribe request.

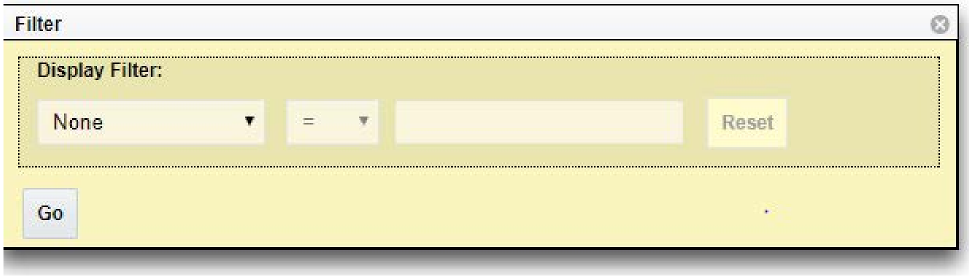
Filtering Ud Client Attribute Map SEC

The filter window allows you to view or filter the Ud Client Attribute data. You can filter the results for specific Ud attributes. The filter provides the results based on the UD Attributes Name, Profile Field Name, and Formatting String.

To filter the Ud Client attribute data:

1. In the Ud Client Attribute Map SEC window, click **Filter**.

   The filter window opens.

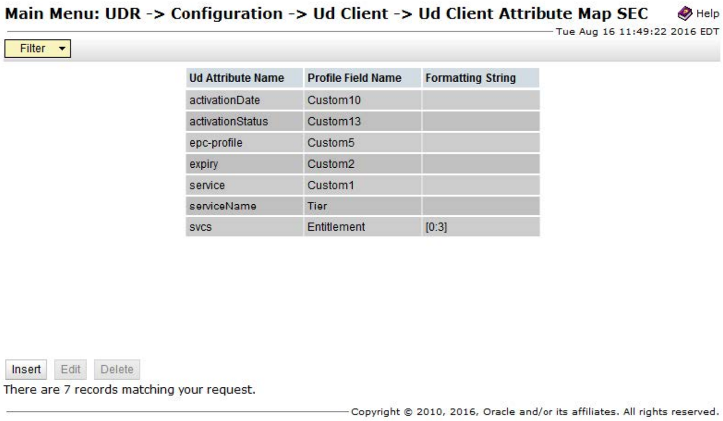


2. From the first list, select the required option from the following:

   - None
   - Ud Attribute Name
   - Profile Field Name
   - Formatting String

   If you selected None, the remaining fields remain disabled and no results are displayed.

3. The second field is auto-populated based on the selection in the first field.
4. (Optional) Enter any value based on the selection in the first field.
5. Click **Go** to view the results or click **Reset** to enter the filter details again.


Main Menu: UDR -> Configuration -> Ud Client -> Ud Client Attribute Map SEC
Help
Tue Aug 16 11:49:22 2016 EDT
Filter
Insert
Edit
Delete

Ud Attribute Name	Profile Field Name	Formatting String
activationDate	Custom10	
activationStatus	Custom13	
epc-profile	Custom5	
expiry	Custom2	
service	Custom1	
serviceName	Tier	
svcs	Entitlement	[0:3]

There are 7 records matching your request.

Inserting Ud Client Attribute Map SEC

You can add a new LDAP attribute mapping to a subscriber Profile field by using insert option in the Ud Client Attribute Map SEC window.

To insert a new LDAP attribute mapping to subscriber profile field:

1. Click **Insert** on the Ud Client Attribute Map SEC window.

   The Insert window opens.

2. In the **Ud Attribute Name** field, enter the Ud attribute name. The maximum number of characters allowed is 64.
3. From the Profile Field Name list, select the profile field name corresponds to the assigned field in the subscriber profile.
4. (Optional) In the **Formatting String** field, enter the formatting string to be applied to retrieve LDAP attribute before assigning a value to subscriber profile field. The maximum number of characters allowed is 64.
5. Click **OK** to add the attribute.

Following table describes the fields for Insert window.

Table 7-3 Insert Window Fields

Field	Description
Ud Attribute Name	Indicates LDAP field associated with subscriber Profile field. By default, the value is not specified. The maximum number of characters allowed is 64.
Profile Field Name	A list provides the profile field name. You can customize the values. By default, the value is not specified.
Formatting String	(Optional) Applied to retrieved LDAP attribute before assigning a value to subscriber Profile field. Formatting String must be a valid regular expression (regex) string. By default, the value is not specified. The maximum number of characters allowed is 64.

Editing Ud Client Attribute Map SEC

You can edit the defined LDAP attribute mapping to a subscriber Profile field by using edit option in the Ud Client Attribute Map SEC window.

To edit an Ud Client Attribute Map SEC field:

1. Click **Edit** on the Ud Client Attribute Map SEC window.

   The Edit window opens.

2. In the Ud Attribute Name field, edit the Ud Attribute name. The maximum number of characters allowed is 64.
3. From the Profile Field Name list, select the profile field name corresponds to the assigned field in the subscriber profile.
4. (Optional) In the Formatting String field, enter the formatting string to be applied to retrieve LDAP attribute before assigning a value to subscriber profile field. The maximum number of characters allowed is 64.
5. Click **OK** to apply the changes.

You have successfully edited the Ud Client Attribute Map SEC.

Deleting Ud Client Attribute Map SEC

You can delete the defined LDAP attribute mapping to a subscriber Profile field by using delete option in the Ud Client Attribute Map SEC window.

To delete an LDAP attribute mapping to subscriber profile field:

1. From the Ud Client Attribute Map SEC window, select the attribute to be deleted.
2. Click **Delete**.

   A confirmation window opens.
3. Click **OK** to delete.

You have successfully deleted the selected Ud Client Attribute Map SEC.

8

Key Performance Indicators for Ud Client

Key performance indicators are added to User Data Repository to measure the performance of the Ud Client on some parameters.

The following table details the KPIs defined for the Ud Client:

Table 8-1 Ud Client Key Performance Indicators

KPI Name	KPI Group	Description
TxUDSearchRate	UDRUD	The number of LDAP Search requests sent per second
TxUdSearchInitialRate	UDRUD	The number of LDAP Search requests sent when initially creating a subscriber sent per second
TxUdSearchReSearchRate	UDRUD	The number of LDAP Search requests sent when performing a research per second
TxUdSubscribeRate	UDRUD	The number of SOAP Subscribe requests sent per second
TxUdSubscribeInitialRate	UDRUD	The number of SOAP Subscribe requests sent when initially creating a subscriber sent per second sent per second
TxUdSubscribeReSubscribeRate	UDRUD	The number of SOAP Subscribe requests sent when performing a re-subscribe per second
RxUdNotifyRate	UDRUD	The number of SOAP Notify requests received per second
RxUdShUdrRate	UDRUD	The number of Sh UDR requests that trigger the Ud-Creation of a subscriber received per second
RxUdShPurRate	UDRUD	The number of Sh PUR requests that trigger the Ud-Creation of a subscriber received per second
RxUdShSnrRate	UDRUD	The number of Sh SNR requests that trigger the Ud-Creation of a subscriber received per second