THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS” WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE. IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

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

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2019 Cisco Systems, Inc. All rights reserved.
Contents

CHAPTER 4  Custom Reference Data Tables  75
  Custom Reference Data Tables  75

CHAPTER 5  Control Center Configurations  79
  Subscribers  79
    Find a Subscriber  79
    Create a Subscriber  80
      General  81
    Edit a Subscriber  85

CHAPTER 6  Notification Services  89
  Notification Services  89
    Apple Push Notifications  89
      Notification Configuration  89
      Message Configuration  91
    GCM Notifications  92
      Notification Configuration  92
      Message Configuration  94
    Service Option Configuration  96

CHAPTER 7  Domain  99
  Domain Overview  99
  Configure Domain for USuM  100
  Configure Domain for External SPR  100
### CHAPTER 8 Services 101

- **Overview** 101
  - **Service** 101
  - **Service Option** 102
  - **Service Configuration** 102
  - **Use Case Template** 102
  - **Use Case Option** 103

- **Service Screens** 103
  - **Default ANDSF Service** 103
  - **Default Location Override ANDSF Service** 108
  - **Location based ANDSF Service** 110
  - **Tier based ANDSF Service** 111
  - **Geo Location based ANDSF Service** 113

- **Configure Lookup for External SPRs** 115
- **Configure Lookup for Extended Device Information** 115
Preface

- About This Guide, on page vii
- Audience, on page vii
- Additional Support, on page vii
- Conventions (all documentation), on page viii
- Communications, Services, and Additional Information, on page ix
- Important Notes, on page ix

About This Guide

This document is a part of the Cisco Policy Suite documentation set.

For information about available documentation, see the CPS Documentation Map for this release at Cisco.com.

Audience

This guide is best used by these readers:

- Network administrators
- Network engineers
- Network operators
- System administrators

This document assumes a general understanding of network architecture, configuration, and operations.

Additional Support

For further documentation and support:

- Contact your Cisco Systems, Inc. technical representative.
- Call the Cisco Systems, Inc. technical support number.
- Write to Cisco Systems, Inc. at support@cisco.com.

## Conventions (all documentation)

This document uses the following conventions.

<table>
<thead>
<tr>
<th>Conventions</th>
<th>Indication</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>bold</strong> font</td>
<td>Commands and keywords and user-entered text appear in <strong>bold</strong> font.</td>
</tr>
<tr>
<td><em>italic</em> font</td>
<td>Document titles, new or emphasized terms, and arguments for which you supply values are in <em>italic</em> font.</td>
</tr>
<tr>
<td>[ ]</td>
<td>Elements in square brackets are optional.</td>
</tr>
<tr>
<td>{x</td>
<td>y</td>
</tr>
<tr>
<td>[ x</td>
<td>y</td>
</tr>
<tr>
<td>string</td>
<td>A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.</td>
</tr>
<tr>
<td>courier font</td>
<td>Terminal sessions and information the system displays appear in courier font.</td>
</tr>
<tr>
<td>&lt; &gt;</td>
<td>Nonprinting characters such as passwords are in angle brackets.</td>
</tr>
<tr>
<td>[ ]</td>
<td>Default responses to system prompts are in square brackets.</td>
</tr>
<tr>
<td>!, #</td>
<td>An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.</td>
</tr>
</tbody>
</table>

---

**Note**

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

---

**Caution**

Means reader be careful. In this situation, you might perform an action that could result in equipment damage or loss of data.
IMPORTANT SAFETY INSTRUCTIONS.

Means danger. You are in a situation that could cause bodily injury. Before you work on any equipment, be aware of the hazards involved with electrical circuitry and be familiar with standard practices for preventing accidents. Use the statement number provided at the end of each warning to locate its translation in the translated safety warnings that accompanied this device.

SAVE THESE INSTRUCTIONS

Note

Regulatory: Provided for additional information and to comply with regulatory and customer requirements.

Communications, Services, and Additional Information

• To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
• To get the business impact you’re looking for with the technologies that matter, visit Cisco Services.
• To submit a service request, visit Cisco Support.
• To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
• To obtain general networking, training, and certification titles, visit Cisco Press.
• To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.

Important Notes

Important

Any feature or GUI functionality that is not documented may not be supported in this release or may be customer specific, and must not be used without consulting your Cisco Account representative.
Overview

The purpose of the Access Network Discovery and Selection Function (ANDSF) is to assist mobile device (User Equipment - UE) to discover non-3GPP access networks such as Wi-Fi or WIMAX that can be used for data communications in addition to 3GPP access networks (such as HSPA or LTE) and to provide the UE with rules policing the connection to these networks.

CPS acts as an ANDSF server i.e. it exposes HTTPS end point so that ANDSF client can POST HTTP request. All request sent to ANDSF server complies to S14 specification and primary use is to provision following data to mobile device (UE)

- Inter-system mobility policy (ISMP) — Used to provision policies related to using access network based on given location and device type (optionally date time)

ANDSF servers can be discovered by UE using DNS based on the MCC and MNC values of HPLMN or VPLMN operator.

ANDSF security will be built first based on HTTPS with server side certificate authentication (less secured) and then will be adding more secured way to authenticate based on USIM (Method Generic Authentication Architecture {3gpp 33.222}).
• UE User Equipment (ANDSF-Client)
• Notification
  • APN Apple Push Notification Server
  • GCM Google Cloud Messaging Server
  • Push Initiator Interface to Push Notification

• UE initiates S14 session with ANDSF server in two cases
  • Server Initiated On receiving notification from GCM/APN cloud.

OR
• UE(Client) Initiated

This s14 session is balanced by policy directors (load balancers) in different Policy Server (QNS) VMs.
During s14 session where package exchange takes place
  • Subscriber information is stored in Subscriber DB.
  • Session Information is stored in Session DB.
  • Policy Retrieval is done through Policy DB.
Client Initiated Session

The numbers represent the traffic flow as below:

- 1 - UE requests to Policy Directors (Load Balancers) in form of DM Packages.
- 2 - Policy Directors (Load Balancers) allocating requests to various Policy Server (QNS) VMs.
- 3/7 - Policy Server (QNS) interacting with policy engine to retrieve requested policies.
- 4/5/6 - Policy Engine interacting with various DBs (Subscriber for subscriber details Session for session details Policy for policy details).
- 8 - Server sending back policy to UE.
Server Initiated Session

WAP Push

Figure 3: WAP Push

The numbers represent the traffic flow as below:

- 1/4 - Server is triggered to check if policy notification is required through various triggers DM MO change SPR change revalidation timer it sends request to policy engine.
- 2/3 - Policy engine interacts with various DBs to reaffirm and if required sends update to Server with notification tokens (GCM/APNS).
- 5 - Server initiates notification to UE by sending notification details along with token to GCM/APNS server.
- 6 - GCM/APNS server sends notification request to UE.
- 7 - UE will then initiate DM session.

Functional Structure

This section covers the following topics:

- Load Balancer
- ANDSF Server
- HA Proxy Configuration
Load Balancer

HAProxy is used for load balancing to fan out HTTP requests to ANDSF endpoints running on CPS VMs. HTTPS end point are configured using server side certificate using HA proxy to expose the HTTPS end point to external client.

ANDSF Server

ANDSF server embeds OMA-DM stack. Server binds to specific IP and port and same is used by HA proxy to load balance the requests. The OMA-DM server is listening for HTTP request, payload of which complies to SyncML Xml (OMA DM spec v1.2). For the ANDSF server, we start an embedded jetty server on Policy Server (QNS) VM listening for HTTP Requests on the port specified.

HA Proxy Configuration

Prerequisite

HAProxy v1.5 is installed on the server.

Steps

Configuring SSL

Generate a key and a certificate using openssl and concatenate them in a file, the certificate first, then the key. To create CA, server and client certificates, perform the following steps:

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Go to directory <code>/etc/pki/tls/certs</code>.</td>
</tr>
</tbody>
</table>
| Step 2 | Create the following directories under certs directory:  

```bash
mkdir Certificates  
mkdir Keys  
mkdir CSR
```
| Step 3 | Create file `index.txt` under certs directory.  

```bash
touch index.txt
```
| Step 4 | Create file `serial` under certs directory.  

```bash
cat "01" > serial
```
| Step 5 | Create a private key of CA  

```bash
openssl genrsa -des3 -out Keys/RootCA.key 2048.
```
| Step 6 | Create self-signed certificate of CA  

```bash
openssl req -config /etc/pki/tls/openssl.cnf -new -x509 -days 360  
-key Keys/RootCA.key -out Certificates/RootCA.crt
```
| Step 7 | Create private key for the server  

```bash
openssl genrsa -des3 -out Keys/server.key 2048.
```
| Step 8 | Create CSR for the server  

```bash
openssl req -config /etc/pki/tls/openssl.cnf -new -key Keys/server.key -out CSR/server.csr
```
| Step 9 | Create server certificate  

```bash
openssl ca -config /etc/pki/tls/openssl.cnf -days 360 -in CSR/server.csr -out Certificates/server.crt -keyfile Keys/RootCA.key -cert Certificates/RootCA.crt -policy policy_anything.
```
Step 10

To remove passphrase from private key so that while starting/restarting the HA proxy, the system does not for passphrase, execute the following commands:

```
cp RootCA.key RootCA.key.orig
openssl rsa -in RootCA.key.orig -out RootCA.key

cp server.key server.key.orig
openssl rsa -in server.key.orig -out server.key
```

Step 11

Copy and paste the RootCA and server certificate and key as certificate first and then the key to a file. For example,

```
vi /etc/pki/tls/certs/haproxy.pem

-----BEGIN CERTIFICATE-----
MIIBozCCABqgCAQgAMIGoGCSqGSIb3DQEBCwUAA4GBASxPpOeBnJzR4S8yvIvBQjA+nGsiTm9wGjM94f4jL2MOuGSpj+K6jxIeEHkCZbI+9EcN4x9b3vah+KZ3sVxwHf0kK9IbVrHMNvR21g8C9mvvMK131+j+8pVv3Lp7d0rC1p9R8JEnWnJWESQac+g-----END CERTIFICATE-----

-----BEGIN RSA PRIVATE KEY-----
MIICXgIBAAKBgQDFxSTUwX5RD4Al2Ya5t5PaAnjcwPa3Km0uaPKSH1USAYmdxc1w8L0k3m9uhSf+KiJsi+TxdfDaYyk/GdDY11C0mm4TM+BLLGAVA2DeNf2h8hBRKbTaGrcXxXOrjQSB/B+1cZ/lq2iq5jzr8xGHZ+t+Rmj2PQFnj1ewlDAQABMA0GCSQGSIb3DQEBAQUAA4GBAB8GULoGWHReSGLY1yAs20uhJ3j/9SvtoueYPBagz5jX4BN0/4hpsKEbCmzytjz7u3v/s0mKoIvVagah77rCZW3kFl6xR67YqZna1ryEB50/MJ9gFC4LNL+sau+W5aIof6+6Ru5N3jnhZST9edJ5GD16/5HTKogvvpw0m-----END RSA PRIVATE KEY-----
```

Step 12

Save and close the haproxy.pem.

Step 13

Edit the haproxy.cfg configuration file in /etc/haproxy/ directory.

Step 14

Go to section, listen controlcenter_proxy and add the following line:

```
listen andsf_proxy
  bind *:8071 ssl crt /etc/pki/tls/certs/haproxy.pem ca-file
  verify optional
```

The section listen andsf_proxy has the endpoint configuration to which UE is sending the requests. It can be modified or optional.

```
default_backend andsf_server
```

The section listen andsf_proxy has the endpoint configuration to which UE is sending the requests. It can be modified or optional.

```
default_backend andsf_server
```

The section listen andsf_proxy has the endpoint configuration to which UE is sending the requests. It can be modified or optional.
server srv1 qns01:8080 check inter 30s
server srv2 qns02:8080 check inter 30s

Here, the received requests are sent to all the server instances in a round-robin manner.

**Note** In a full blade setup, the above configuration needs to be done in both primary and secondary blades and the files `/etc/pki/tls/certs/haproxy.pem` and `/etc/pki/tls/certs/Certificates/RootCA.crt` must be present in both the blades.

---

## Use Cases

The following table provides a list of use cases:

**Table 1: Use Cases**

<table>
<thead>
<tr>
<th>Cisco Use Case</th>
<th>Description</th>
<th>Device Type Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wi-Fi Link Quality</td>
<td>Server will push policy to connect to SSID.</td>
<td>Android/iOS</td>
</tr>
<tr>
<td></td>
<td>Note We will need custom MO to supply the managed and non-manage network.</td>
<td></td>
</tr>
<tr>
<td>Location Based Access</td>
<td>ANDSF server will receive UE Location from client and send Policy MOs specific to the location.</td>
<td>Android/iOS</td>
</tr>
<tr>
<td>Subscriber Profiling</td>
<td>ANDSF server will receive IMSI and based on the UE's profile pushes policy MOs.</td>
<td>Android/iOS</td>
</tr>
<tr>
<td>Location and Subscriber Profiling</td>
<td>ANDSF server will receive IMSI and based on the UE's profile and server location policies pushes policy MOs.</td>
<td>Android/iOS</td>
</tr>
<tr>
<td>Power Geo-Fence</td>
<td>Policy Selection (based on Loc/Cell-ID; No Long/Lat). Update Policy flag will be set to True. Server determines the 3GPP_Locations where service shall be ON.</td>
<td>Android/iOS</td>
</tr>
<tr>
<td>Time Based Access</td>
<td>Generic requirement Policy selection filtered based on location and subscriber profile.</td>
<td>Android/iOS</td>
</tr>
<tr>
<td>Cisco Use Case</td>
<td>Description</td>
<td>Device Type Support</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>Wi-Fi Link Quality</td>
<td>Client uses QoS parameter defined under MO Ext to identify allowed QoS and signal strength before switching to a specific SSID. This use case is supported only for Android.</td>
<td>Android</td>
</tr>
</tbody>
</table>
| Server Triggered Push Policy   | Apple/GCM push notification will be sent for following scenarios  
• Any SPR profile change (SSID Tier value etc) If SPR database changed for any value it will trigger notification  
• Revalidation Timer Expiry  
• Policy-MO change will also trigger the notification                                                                                     | Android/iOS        |

**Call Flows**

The following call flows are covered in this section:

- **Android Pull**
- **Android Push**
- **iPhone Pull**
- **iPhone Push**
Android Pull

Figure 4: Call Flow - Pull and ANDSF Server Request for GCM Token

[1] Perform Network Selection

[2] Package#1: DevInfo, Alert 1201

[3] Package#2: Get for IMSI, GCMToken and UE-Location

[4] Package#3: Result for IMSI, GCMToken and UE-Location

[5] Package#4: Replace ANDSF policy rules

[6] Package#3: Status

[7] Package#4: End-of-Session

Client must send the GCM Token as part of the PULL procedure to ensure that the ANDSF Server always has a valid GCM Token.

Android Push

Figure 5: GCM Notification and Policy Update (PUSH)
**iPhone Pull**

*Figure 6: iPhone Call Flow*

- **Package#1**: DevInfo containing UUID
- **Package#2**: Asks for location
- **Package#3**: UE provides location (lat and long)
- **Server calculates policy based on location received**
- **Package#4**: Policy to UE
- **Package#5**: Ack of policy received
- **Package#6**: Final package to close OMA DM package exchange

*Policy would contain .mobileconfig and .cert file together with other attributes*

*[S14 call flow ends here]*

**http GET for *.mobileconfig file**

- Responds with *.mobileconfig file
- **http GET for *.cert file**
- Responds with *.cert file

**Prompts subscriber for further installation**

**iPhone Push**

*Figure 7: APNS Call Flow*

- **UE**
  - **Initiate registration**
  - Returns APNS registration token
  - Provides APNS token in Package#1

- **Further package exchange to complete the pull scenario (client initiated)**

- **Delivers APNS message**
  - UE initiates ANDSF session by sending Package#1
  - Further package exchange to get policy (server initiated)

- **APNS Server**
  - Sends APNS notification (providing APNS token)

- **ANDSF**
Overview

In ANDSF, reference data is considered information that is needed to operate the policy engine, but not used for evaluating policies. For example, in the Reference Data tab in Cisco Policy Builder, are the forms used to define systems, clusters, and instances, and to set times and dates used for tariff switching. The policy engine needs to refer to this data only to process policies correctly. However, the data does not define the policy itself.

Policy Builder provides core plug-ins for customizing and optimizing your installation.

- Configurations set at the system level are system-wide except as noted in the bullet items below.
- Configurations set at the cluster level apply to that cluster and the instances in it. A value set here overrides the same value set at the system level.
- Configurations set at the instance level apply to the instance only and override the same value set at the cluster or system level.

Select the Create Child action in a Plug-in Configuration node in the Systems tree to define them. You can change any of the variables from the default, or choose not to use a plug-in, as necessary.

When you create a system from the example, the following configuration stubs appear at the cluster and instance level.
Figure 8: Plugin Configurations Summary

A threading configuration utility is provided for advanced users and future development.

Click **Threading Configuration** from right pane to add the threading configuration to the system. If you are planning to run the system with higher TPS then you need to configure Threading Configuration. For further information contact your Cisco Technical Representative.

An example configuration is shown below:

Figure 9: Threading Configuration

The following parameters can be configured under Threading Configuration:
Table 2: Threading Configuration Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thread Pool Name</td>
<td>Name of the Cisco thread pool.</td>
</tr>
<tr>
<td>Threads</td>
<td>Threads to set in the thread pool. You can set Rules Thread to 50/100</td>
</tr>
<tr>
<td></td>
<td>depending on call flow (based on number of lookup and per transaction</td>
</tr>
<tr>
<td></td>
<td>round trip time).</td>
</tr>
<tr>
<td></td>
<td>• rules = 50; Queue Size = 0; Scale By Cpu Core = unchecked</td>
</tr>
<tr>
<td></td>
<td>• rules = 100; Queue Size = 0 (If TPS is &gt; 2000 per Policy Server (QNS)</td>
</tr>
<tr>
<td></td>
<td>depending on call model used, for example, if LDAP is enabled); Scale</td>
</tr>
<tr>
<td></td>
<td>By Cpu core = unchecked</td>
</tr>
<tr>
<td>Queue Size</td>
<td>Size of the queue before they are rejected.</td>
</tr>
<tr>
<td>Scale By Cpu Core</td>
<td>Select this check box to scale the maximum number of threads by the</td>
</tr>
<tr>
<td></td>
<td>processor cores.</td>
</tr>
</tbody>
</table>

Async Threading Configuration

You are always required to select this configuration but no changes to it are necessary. Click Async Threading Configuration from right pane to add the configuration in the system.

Use the default values for the Async Threading Plugin. For further information contact your Cisco Technical Representative.

Note

Always select the link for Async Threading Configuration to configure your ANDSF system.
The following parameters can be configured under Async Threading Configuration:

**Table 3: Async Threading Configuration Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Processing Threads</td>
<td>The number of threads that are allocated to process actions based on priority.</td>
</tr>
<tr>
<td>Default Action Priority</td>
<td>The priority assigned to an action if it is not specified in the Action Configurations table.</td>
</tr>
<tr>
<td>Default Action Threads</td>
<td>The number of threads assigned to process the action if it is not specified in the Action Configurations table.</td>
</tr>
<tr>
<td>Default Action Queue Size</td>
<td>The number of actions that can be queued up for an action if it is not specified in the Action Configurations table.</td>
</tr>
<tr>
<td>Default Action Drop Oldest When Full</td>
<td>When checked the oldest queued action is dropped from the queue when a new action is added to a full queue. Otherwise the new action to add is ignored. This check box applies to all the threads specified in the fields above. To drop a specific thread leave this unchecked and use the Action Configurations table.</td>
</tr>
<tr>
<td>Action Configurations Table</td>
<td>N/A</td>
</tr>
<tr>
<td>Action Name</td>
<td>The name of the action. This must match the implementation class name.</td>
</tr>
</tbody>
</table>
### Custom Reference Data Configuration

Before you can create a Custom reference data table configure your system to use the Custom Reference Data Table plug-in configuration.

You only have to do this one time for each system cluster or instance. Then you can create as many tables as needed.

Click **Custom Reference Data Configuration** from right pane to add the configuration in the system.

Here is an example:

- **HA example:**
  - Primary Database IP Address `sessionmgr01`
  - Secondary Database IP Address `sessionmgr02`
  - Database Port `27717`

- **AIO example:**
  - Primary Database IP Address localhost or `127.0.0.1`
  - Secondary Database IP Address NA (leave blank)
  - Database Port `27017`

The following parameters can be configured under Custom Reference Data Configuration.

**Table 4: Custom Reference Data Configuration Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary Database IP Address</td>
<td>IP address of the primary sessionmgr database.</td>
</tr>
<tr>
<td>Secondary Database IP Address</td>
<td>Optional this field is the IP address of a secondary backup or failover sessionmgr database.</td>
</tr>
</tbody>
</table>
### Unified API Configuration

Click **Unified API Configuration** from right pane to add the configuration in the system.

*Figure 11: Unified API Configuration*

The following parameters can be configured under Unified API Configuration.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Port</td>
<td>Port number of the sessionmgr. It should be the same for both the primary and secondary databases.</td>
</tr>
<tr>
<td>Db Read Preference</td>
<td>Read preference describes how sessionmgr clients route read operations to members of a replica set. You can select from the following drop-down list:</td>
</tr>
<tr>
<td></td>
<td>• Primary Default mode. All operations read from the current replica set primary.</td>
</tr>
<tr>
<td></td>
<td>• PrimaryPreferred In most situations operations read from the primary but if it is unavailable operations read from secondary members.</td>
</tr>
<tr>
<td></td>
<td>• Secondary All operations read from the secondary members of the replica set.</td>
</tr>
<tr>
<td></td>
<td>• SecondaryPreferred In most situations operations read from secondary members but if no secondary members are available operations read from the primary.</td>
</tr>
<tr>
<td></td>
<td>For more information refer to <a href="http://docs.mongodb.org/manual/core/read-preference/">http://docs.mongodb.org/manual/core/read-preference/</a>.</td>
</tr>
<tr>
<td>Connection Per Host</td>
<td>Number of connections that are allowed per DB Host. Default value is 100.</td>
</tr>
</tbody>
</table>
Table 5: Unified API Configuration

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fields To Wrap With Cdata Tags</td>
<td>This is a CSV separated string. The Unified API now can handle CDATA fields. Use the Plug-in configuration in Policy Builder to set CDATA fields for the main Unified API. The property ua.cdata.fields is used to set the fields that should be wrapped in CDATA tags for the client CommFactory to properly send and receive API requests. -Dua.cdata.fields=networkIdpassworddataoldNetworkIdoldPasswordnewPassword is the default.</td>
</tr>
<tr>
<td>Submit Requests To Audit Log</td>
<td>Select the check-box to log requests to API in audit log. Default is True (checked).</td>
</tr>
<tr>
<td>Submit Read Requests To Audit Log</td>
<td>Select this check-box to log read requests in audit log. Default is False (unchecked).</td>
</tr>
</tbody>
</table>

Notification Configuration

Currently, Cisco Policy Builder offers following notification types for ANDSF:

- Apple iOS devices/iPhone® push (iOS devices)
- Google Cloud Messaging

Click Notification Configuration from right pane to add the configuration in the system.

![Notification Configuration](image)

The following parameters can be configured under Notification Configuration.

Table 6: Notification Configuration Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple Push Notification Configuration</td>
<td>Select this check box to configure the connection for a push to an Apple iOS device or iPhone. For more information, refer to Notification Services, on page 89.</td>
</tr>
<tr>
<td>Google Cloud Messaging Configuration</td>
<td>Select this check box to configure the connection for a google cloud messaging notification. For more information, refer to Notification Services, on page 89.</td>
</tr>
</tbody>
</table>
USuM Configuration

Click **USuM Configuration** from right pane to add the configuration in the system.

*Figure 13: USuM Configuration*

The following parameters can be configured in Spr Configuration pane under USuM Configuration:

*Table 7: USuM Configuration Parameters*

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Spr Configuration</strong></td>
<td></td>
</tr>
<tr>
<td>Disable Regex Search</td>
<td>Not applicable leave to default.</td>
</tr>
<tr>
<td>Enable Avp Regex Search</td>
<td>Not applicable leave to default.</td>
</tr>
<tr>
<td>Exclude Suspended Subscribers From Policy</td>
<td>In case of subscriber state is Suspended SPR will not validate IMSI.</td>
</tr>
<tr>
<td><strong>Search Query Results Limit</strong></td>
<td>Used to limit search if we are not passing any IMSI/MSISDN (NetworkID) in control center to list subscriber. Default value is 1000.</td>
</tr>
<tr>
<td><strong>Max Number Of Locations To Store In History</strong></td>
<td>Not applicable leave to default.</td>
</tr>
<tr>
<td><strong>Last Visited Date Threshold</strong></td>
<td>Not applicable leave to default.</td>
</tr>
</tbody>
</table>
The following parameters can be configured in Policy Engine Submission Configuration pane under USuM Configuration:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable check box</td>
<td>Leave it to default.</td>
</tr>
<tr>
<td>Message Queue Size</td>
<td>Queue to hold data to generate internal SPR Refresh events for policy engine during Create Update Delete of subscriber.</td>
</tr>
<tr>
<td>Message Queue Sleep</td>
<td>Sleep before popping next batch for generating SPR Refresh events for policy engine for RAR processing.</td>
</tr>
<tr>
<td>Message Queue Batch Size</td>
<td>Batch size for fetching number of subscriberIds in one go for generating SPR Refresh events for policy engine for RAR processing.</td>
</tr>
<tr>
<td>Message Queue Pool Size</td>
<td>Message queue pool size to consume the data from queue and generate SPR Refresh events.</td>
</tr>
<tr>
<td>Notification Rate Limit</td>
<td>Rate limiting for generating SPR Refresh events. SPR Refresh events is used to generate RAR for active session where subscriber data has been change.</td>
</tr>
</tbody>
</table>
Figure 15: Database Configuration

*Database Configuration

- **Use Minimum Indexes**

*Db Write Concern*

- **OneInstanceSafe**

*Db Read Preference*

- **Secondary**

*Failover Sla Ms*

- **2000**

*Max Replication Wait Time Ms*

- **100**

*Shard Configuration*

*Primary Database Host*

- **site1-sessionmgr01**

Secondary Database Host

- **site1-sessionmgr02**

*Database Port*

- **27720**

Remote Shard Configuration

*Tertiary Database Host*

- **site2-sessionmgr01**

Quaternary Database Host

- **site2-sessionmgr02**

The following parameters can be configured in Database Configuration pane under USuM Configuration:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Configuration</td>
<td></td>
</tr>
</tbody>
</table>
### Parameter | Description
---|---
Use Minimum Indexes | It is used to decide what all indexes need to be created on SPR collection by default and need all the indexes to be created (Check this when subscriber is very low e.g. < 50K). Default value is unchecked.
Db Write Concern | Controls the write behavior of sessionmgr and for what errors exceptions are raised. Default option is OneInstanceSafe.
Db Read Preference | Read preference describes how sessionmgr clients route read operations to members of a replica set. You can select from the following drop-down list
- Primary
- PrimaryPreferred
- Secondary
- SecondaryPreferred
For more information refer to [http://docs.mongodb.org/manual/core/read-preference/](http://docs.mongodb.org/manual/core/read-preference/).
Failover Sla Ms | This parameter is used to enter the amount of time to wait before starting failover database handling. The time is in milliseconds.
Max Replication Wait Time Ms | This option specifies a time limit in milliseconds for the write concern. This parameter is applicable only if you select TwoInstanceSafe in Db Write Concern.
This parameter causes write operations to return with an error after the specified limit even if the required write concern eventually succeeds. When these write operations return MongoDB does not undo successful data modifications performed before the write concern exceeded the replication wait time limit. This time is in milliseconds.

**Shard Configuration**

**Important** The host names must exactly be the same host name used when the corresponding replica-set is created in Mongo. Only the data holding members need to be configured (and not the arbiters).

| Primary Database Host | String - Primary Host Address. |
| Secondary Database Host | String - Secondary Host Address. |
| Database Port | Default value is 27720. |

**Remote Shard Configuration**

**Important** Remote shard configuration is used only for GR deployments. The host names must exactly be the same host name used when the corresponding replica-set is created in Mongo. Only the data holding members need to be configured (and not the arbiters).

| Tertiary Database Host | String - Tertiary Host Address. |
Click Add to add a new row on Remote Database Configuration pane. The following parameters can be configured in Remote Database Configuration pane under USuM Configuration:

**Table 10: Remote Database Configuration Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>String - Name of the remote database.</td>
</tr>
<tr>
<td>Key Prefix</td>
<td>Key prefix to be match for the remote database to be selected for lookup.</td>
</tr>
<tr>
<td>Connections Per Host</td>
<td>Number of connections that can be created per host. Default value is 5.</td>
</tr>
<tr>
<td>Db Read Preference</td>
<td>Read preference describes how sessionmgr clients route read operations to members of a replica set. You can select from the following drop-down list</td>
</tr>
<tr>
<td></td>
<td>• Primary</td>
</tr>
<tr>
<td></td>
<td>• PrimaryPreferred</td>
</tr>
<tr>
<td></td>
<td>• Secondary</td>
</tr>
<tr>
<td></td>
<td>• SecondaryPreferred</td>
</tr>
<tr>
<td>Primary Database Host</td>
<td>Host name of the remote sessionmgr database.</td>
</tr>
<tr>
<td>Secondary Database Host</td>
<td>(Optional) Host name of a secondary, backup, or failover sessionmgr database.</td>
</tr>
<tr>
<td>Tertiary Database Host</td>
<td>Host name of the tertiary database.</td>
</tr>
<tr>
<td>Quaternary Database Host</td>
<td>Host name of the quaternary database.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>Port</td>
<td>Port number of the remote sessionmgr database. It must be the same for both the primary and secondary databases. Default value is 27720.</td>
</tr>
</tbody>
</table>

**REST API Configuration**

ANDSF will implement the http/https protocol to receive messages using the REST style of request/response structure. XML strings will be sent in the payload.

Click **REST API Configuration** from right pane to add the configuration in the system.

*Figure 17: REST API Configuration*

The following parameters can be configured under REST API Configuration:
### Table 11: REST API Configuration Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fields To Wrap With Cdata Tags</td>
<td>Here we give the SyncML fields which will be wrapped by server in CDATA tag.</td>
</tr>
<tr>
<td></td>
<td>In an XML document or external parsed entity a CDATA section is a section of element content that is marked for the parser to interpret as only character data not markup. A CDATA section is merely an alternative syntax for expressing character data; there is no semantic difference between character data that manifests as a CDATA section and character data that manifests as in the usual syntax in which “&lt;” and “&amp;” would be represented by “&lt;” and “&amp;,” respectively. A CDATA section starts with the following sequence <code>&lt;![CDATA[ and ends with the first occurrence of the sequence ]]&gt;</code> There are two types of REST API requests that come from client Read and Write.</td>
</tr>
<tr>
<td></td>
<td>• Read Requests are generally used for retrieving data from Server using REST API. Examples of these requests are http://${#Project#ServiceEndpoint}/qps/rest/andsf/static/policy?&lt;name_of_mobile_config&gt;.mobileconfig. This is to get mobileconfig</td>
</tr>
<tr>
<td></td>
<td>• Similarly Write Requests are used for updating data in server using REST API.</td>
</tr>
<tr>
<td>Keep Alive Custom response headers (provide multiple headers separated with comma)</td>
<td>Here we can attach custom response headers along with 2000 K whenever keep-alive requests come from client.</td>
</tr>
<tr>
<td>Submit Requests To Audit Log</td>
<td>Checking this will audit all (read and write requests) for auditing.</td>
</tr>
<tr>
<td>Submit Read Requests To Audit Log</td>
<td>Checking this will audit only read requests for auditing.</td>
</tr>
<tr>
<td>Validate Requests Against Scheme</td>
<td>Checking this will validate client requests according to ANDSF schema and not return response if there is schema invalidation.</td>
</tr>
<tr>
<td>Json Payload</td>
<td>Checking this will enable payload to be parsed as JSON instead of XML which is currently the default markup language.</td>
</tr>
</tbody>
</table>

### ANDSF Configuration

Click **ANDSF Configuration** from right pane to add the configuration in the system.

The following parameters can be configured under ANDSF Configuration:
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Max URIs in each MO in Session                                  | It indicates the number the URI with values for each MO sent to be stored in the session in DB. Default value is 5. When we respond back to the client with policies we respond with elements we have configured in DM Configuration. These elements are stored as MOs and each MO has certain URIs in a particular session. The value in this box is a delimiter to the number of URIs that is stored for each MO. Example of a URI:  
  \(<X>/Policy/<X>/ValidityArea/WLAN_{Location}/<X>/HESSID>\)  
  \(<X>/Policy/<X>/ValidityArea/WLAN_{Location}/<X>/SSID\)  
  \(<X>/Policy/<X>/ValidityArea/WLAN_{Location}/<X>/BSSID\)                                                                                     |
| Re-validation Timer (in Minutes)                                | The time when the server will start generating notifications for sessions which are idle upto those number of minutes to check for stale sessions.                                                                                                                                  |
| Enable URI Format for Request/Response                          | Enabling this check box will enable URI format of data exchange in both request and response. For more information, refer to Enable URI Format for Request/Response, on page 26.                                                                                                       |
| Max GeoFence Distance For GeoQuery (in meters)                  | Related to Apple Devices (for geo-location) It indicates the distance with which we query for the GeoFence circles from the input latitude and longitude coordinates.                                                                                                           |
| Directory containing static policy files                        | (Related to Apple Devices) In order to switch from a WLAN to TGPP network and vice versa apple devices need certain configurations (mobileconfig) to perform that. This path specifies the directory where this and other related files are stored and are sent to client. |
| File Extension To Http Headers Map                              | (Related to Apple Devices) This is http headers e.g. when iOS mobile-config and certificates are sent to UE (ANDSF-Server) over https these are the additional http-headers included in the http response.                                                                                          |
| Extension                                                       |                                                                                               |
| Http Headers                                                   |                                                                                               |
| Is Binary File                                                 |                                                                                               |
| External SPR Url                                                | ANDSF uses the external SPR URL to validate the subscriber. Policy Server provides a policy for the valid subscribers. If you are using an external SPR for authorization, be sure to configure the domain as described in Domain Overview, on page 99. |
Enable URI Format for Request/Response

URI Format:

```
<Item>
  <Source>
    <LocURI>./UE_Location/Geo_Location/Latitude</LocURI>
  </Source>
  <Data>110001110101000101101000</Data>
</Item>
<Item>
  <Source>
    <LocURI>./UE_Location/Geo_Location/Longitude</LocURI>
  </Source>
  <Data>101000001001111100110010</Data>
</Item>
```

Default XML Format:

```
<Item>
  <Source>
    <LocURI>./UE_Location</LocURI>
  </Source>
  <Data>![CDATA[<UE_Location>
    <Geo_Location>
      <Latitude>110001110101000101101000</Latitude>
      <Longitude>101000001001111100110010</Longitude>
    </Geo_Location>
  ]]></UE_Location>
</Data>
```

For description on parameters related to Database Configuration and Shard Configuration refer to USuM Configuration, on page 18.

Based on the configuration in ANDSF Policy Audit ANDSF network device manager collects the copy of the policies which are to be delivered to the UE and store it in audit collection of ANDSF database. In policy builder if the enable audit is not selected then audit collection won't be performed at network device manager. When the size of the collection reaches the capped collection size then old documents are automatically purged.

**Figure 18: ANDSF Policy Audit Configuration**

The following parameters can be configured under ANDSF Policy Audit Configuration:

- **Enable Audit**
- **Capped Collection Size (In GBs)**: 0.1
Table 13: ANDSF Policy Audit Configuration Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable Audit</td>
<td>Select this check box to enable auditing information (internal and external) related to ANDSF.</td>
</tr>
<tr>
<td>Capped Collection Size (In GBs)</td>
<td>Size of audit information to be stored.</td>
</tr>
</tbody>
</table>
DM Configuration

- In Policy Builder, on page 29
- In Control Center, on page 42
- Using REST APIs, on page 63

In Policy Builder

In this chapter we have described DM Configuration both in Policy Builder and Control Center.

Note

Any DM configuration changes must be done in consultation with your Cisco Technical Representative.

Definition of template according to specifications, i.e. it contains some read only templates. We can also create our own templates to be used in control center.

DM Trees

These DM Trees define set of pre-configured rules that prioritize the access network when the UE is not capable to connect to the network through multiple accesses. Every node is in correspondence to specifications, coupled in four DM Trees with base ISMP (Policy Tree) having most common nodes.

Policy Tree (ISMP)

This policy tree consists of basic nodes:

- Access Technology Information
- Location Based Information: TGPP, WLAN and Geo Location
- Time of Day Based Information: Time/Date; Start/Stop
In addition to the above mentioned parameters, the following parameters can be configured:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the DM tree template.</td>
</tr>
<tr>
<td>MO Type</td>
<td>It defines the type of DM Tree.</td>
</tr>
<tr>
<td>Hide in Control Center</td>
<td>If we select this check box, this DM Tree will not be visible in Control Center. Default is unchecked.</td>
</tr>
<tr>
<td>Copy Current DM Tree</td>
<td>Selecting this will make a copy of the selected (current) DM Tree.</td>
</tr>
<tr>
<td>Nodes</td>
<td>This defines structure of the DM Tree.</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the node.</td>
</tr>
<tr>
<td>Display Name</td>
<td>Display name of Node will be used to show node in control center.</td>
</tr>
<tr>
<td>Type</td>
<td>Type can be Node or Element.</td>
</tr>
<tr>
<td>Element Type</td>
<td>If node is Element, Element type can be Int, Text, Binary, Boolean.</td>
</tr>
<tr>
<td>Parent Node</td>
<td>Parent node of this node.</td>
</tr>
<tr>
<td>Mandatory</td>
<td>If one instance is must this needs to be true else false.</td>
</tr>
</tbody>
</table>
**Parameter** | **Description**
---|---
Node Relation | Node relation is relation of this node with its parent. It can be One-to-One or One-to-Many.
Upper Bound | If node relation is One-to-many, upper bound specifies maximum number of instance that can be created. 0 mean unlimited.
Valid Values | Depending on the node type, valid values can be specified. For example, if node type is boolean, then valid value will be True or False.
- All - All values
- List of Valid Values - List of values can be provided as valid values. This is applicable only for elements.

**Policy Tree Extension (ISMP_EXT)**

In addition to all the basic nodes of Policy Tree, this tree contains an additional node: Ext. This Ext node further provides additional location and TimeofDay based nodes, as well as QoS related nodes.

*Figure 20: Policy Tree Extension (ISMP_EXT)*

**Policy Default Location Override (ISMP_DEF_LOC_OVERRIDE)**

In addition to all the basic nodes of Policy Tree Extension, this tree contains following two additional nodes:

- UE_Location_TGPP_Location
- UE_Location_WLAN_Location

Adding these nodes, makes this tree override TGPP Location or WLAN location with incoming TGPP/WLAN location.
Figure 21: Policy Default Location Override (ISMP_DEF_LOC_OVERRIDE)

The tree consists of only TGPP and WLAN location based nodes.

Figure 22: UE_Location

This tree consists of only TGPP and WLAN location based nodes.

UE_Location

In addition to the above mentioned read only templates, the user can create their own customized templates. An example configuration (DM Tree for GEO_Location) is shown below:

Geo_Location

In addition to the above mentioned read only templates, the user can create their own customized templates. An example configuration (DM Tree for GEO_Location) is shown below:
DM Tree Lookups

In DM Tree Lookups, we define the template of what keys we are supposed retrieve a policy for a DM tree. There are some Read Only templates already defined under it for the convenience of the customer.

Policy Extension Default (Policy_DEF_OVERRIDE)

This lookup is the default lookup, which retrieves policy without seeking any incoming information.
The following are various configuration parameters:

**Table 15: DM Tree Lookup Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the lookup template.</td>
</tr>
<tr>
<td>DM Tree</td>
<td>It defines the type of DM Tree for which lookup is configured.</td>
</tr>
<tr>
<td>Copy Current DM Tree Lookup</td>
<td>Selecting this will make a copy of the selected (current) DM Tree lookup.</td>
</tr>
<tr>
<td>Keys</td>
<td>This defines the search criteria structure of the lookup.</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the node.</td>
</tr>
<tr>
<td>Display Name</td>
<td>Display name of Node will be used to show node in control center.</td>
</tr>
<tr>
<td>Is Mandatory</td>
<td>If one instance is must this needs to be true else false.</td>
</tr>
<tr>
<td>Type</td>
<td>Currently, we support Text only.</td>
</tr>
</tbody>
</table>

**Figure 24: Policy Extension Default (Policy_DEF_OVERRIDE)**
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valid Values</td>
<td>We can specify what will be the valid values that can be accepted for the current DM Tree Lookup. We can specify our own valid values by importing the valid ones from another tables column.</td>
</tr>
</tbody>
</table>
| Runtime Binding   | This is used to dynamically bind and match key values coming at runtime from the UE with the values in database (CRD table). Each Key can be bind to following attributes:  
  • None: No dynamic binding  
  • Bind to Subscriber AVP code: Match DB value with one coming in `<avp:value>` for the `<avp:code>` we enter here. Subscriber AVP (custom data) can be used to get the value. For example, tier  
  • Bind to Session/Policy State Field: Policy State Retrievers can be configured to get the value, like Session IMSI, Session UserName.  
  • Bind to DM Request URI: Value can also be retrieve from the DM Request URI. Match DB value with one coming in a particular URI of the DM request package. For example, `/UE_Location/WLAN_Location`. |

**Policy Extension Location (Policy_EXT_Location)**

This lookup is used for fetching only incoming location based information, for android devices (TGPP and WLAN Location).
Policy Extension Tier (Policy_EXT_TIER)

This lookup is used for fetching tier based information for a particular subscriber. This value is bound to its Subscriber information in control center. Furthermore, it also fetches location based information, i.e. TGPP and WLAN locations.
**Geo Location (Policy_EXT_GEO_LOC_STATIC)**

In addition to the read only templates, the user can create their own customized templates. An example configuration (DMT Lookup for GEO_Location) is shown below. This lookup example is used for fetching incoming geo_location for apple based devices.

---

**Figure 26: Policy Extension Tier (Policy_EXT_TIER)**

<table>
<thead>
<tr>
<th>DM Tree Lookup (Read Only)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td>Policy_EXT_TIER</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

**Keys**

<table>
<thead>
<tr>
<th>Name</th>
<th>Display Name</th>
<th>Is Mandatory</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>TGPP_Location</td>
<td>TGPP_Location</td>
<td>✔</td>
<td>Text</td>
</tr>
<tr>
<td>WLAN_Location</td>
<td>WLAN_Location</td>
<td>✔</td>
<td>Text</td>
</tr>
<tr>
<td>Tier</td>
<td>Tier</td>
<td>✔</td>
<td>Text</td>
</tr>
</tbody>
</table>

**Valid Values**

Valid values pulled from another table's column (key)

**Runtime Binding**

Which templates match when a message is received

- None
- Bind to Subscriber AVP code
- Bind to Session/Policy State Field
- Bind to DM Request URI

*NOTE: Keys cannot be changed after publishing. This is to protect integrity of existing lookups. Keys can always be added or deleted.*
Figure 27: Geo Location (Policy_EXT_GEO_LOC_STATIC)

<table>
<thead>
<tr>
<th>Name</th>
<th>URI Type</th>
<th>Display Name</th>
<th>Is Mandatory</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Policy_EXT_GEO_LOC_STATIC</td>
<td></td>
<td>GEO_Location</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Some or all keys in this lookup have been published and will be read only. Newly added keys will be editable.

Valid Values
- Valid values pulled from another table's column (key)

Runtime Binding
- Which templates match when a message is received
  - None
  - Bind to Subscriber AVP code
  - Bind to Session/Policy State Field
  - Bind to DM Request URI

Key Details

NOTE: Keys cannot be changed after publishing. This is to protect integrity of existing lookups. Keys can always be added or deleted.

URI Types

URI type defines the URIs that server can receive from the client device. It is used to bind with DM Tree lookup key. There are some Read Only templates already defined under it for the convenience of the customer.

DevInfo

This defines device information URI Type, which needs to be stored in ANDSF Client.
The following are various configuration parameters:

Table 16: URI Type Parameters - DevInfo

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of URI type. For example, UE_Location from DM Tree Lookup Parameters.</td>
</tr>
<tr>
<td>Schema</td>
<td>UE_Location (from CRD) from DM Tree Lookup Parameters.</td>
</tr>
<tr>
<td>Get Request URI</td>
<td>This URI will be used by server to send Get command to client. For example, /UE_Location (URI which is looked up in UE Request) from DM Tree Lookup Parameters.</td>
</tr>
<tr>
<td>Alert URN</td>
<td>If data is coming in alert with urn this will be used to map that data with this URI type.</td>
</tr>
<tr>
<td>Copy Current URI Type</td>
<td>Selecting this will make a copy of the selected (current) URI Type.</td>
</tr>
<tr>
<td>URI List</td>
<td>This table define the URI that server will read from the request. Corresponding custom reference data table will be used to find all fields to read.</td>
</tr>
</tbody>
</table>

**UELocation**

This defines equipment location URI Type, for android devices, in the form of TGPP and WLAN locations. These are stored in TGPP and WLAN reference tables.
In addition to the read-only templates, the user can create their own customized templates. An example configuration is shown below. This example defines equipment location URI Type, for apple devices, in the form of Geo location. This is stored in Geo_Location reference table.

**Figure 30: URI Type - GEO Location**

<table>
<thead>
<tr>
<th>Name</th>
<th>Schema</th>
</tr>
</thead>
<tbody>
<tr>
<td>UELocation GEO LOC</td>
<td>UELocation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>URI List</th>
</tr>
</thead>
<tbody>
<tr>
<td>/UELocation/GEOLocation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>URI</th>
<th>Alert URN</th>
</tr>
</thead>
<tbody>
<tr>
<td>/UELocation/GEOLocation</td>
<td>unicom:at-ext-3gpp-andsf:1.0</td>
</tr>
</tbody>
</table>

**GEO Location**

In addition to the read only templates, the user can create their own customized templates. An example configuration is shown below. This example defines equipment location URI Type, for apple devices, in the form of Geo location. This is stored in Geo_Location reference table.
DM Client Vendors

In DM Client Vendors we define client vendor related information along with notification token related information.

Figure 31: DM Client Vendor

The following parameters can be configured under DM Client Vendor:

Table 17: DM Client Vendor Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of Client Vendor.</td>
</tr>
<tr>
<td>Client Name Tag (.DevInfo)</td>
<td>Name of tag in UE xml. For example &lt;UEClientVendor&gt; in UE Request. To be written as Ext_UERequestVendor in this field.</td>
</tr>
<tr>
<td>Client Name Value</td>
<td>Value inside tag of UE xml. For example &lt;Ext&gt;&lt;UEClientVendor&gt;SmartSwitch&lt;/UEClientVendor&gt;&lt;/Ext&gt; hence value here will be SmartSwitch.</td>
</tr>
<tr>
<td>DM Root URI (Prefix)</td>
<td>/ANDSF</td>
</tr>
<tr>
<td>ISMP Rule Priority URI</td>
<td>/Policy/X+/RulePriority</td>
</tr>
</tbody>
</table>
In Control Center

In Control Center, under DM Configuration tab, the user can see all the windows corresponding to DM Configuration defined in Policy Builder.

DM Trees

Below window shows the list of DM trees. User can add the new DM trees by clicking on the button add and providing the name for the tree as shown below.

*Figure 32: DM Trees - ISMP*

While adding the tree user can select individual nodes and provide the values for the elements as shown below.
You need to add the following details:

1. Need to give DM Tree Name.
2. Give policy name. Over here you can give priority to your Policy.
3. Prioritized Access You need to perform the following under Prioritized Access
   - Give a name to Access.
   - Select Access Technology.
   - Put accessId and Secondary AccessId.

   Basically here we are configuring policies based on a DM structure and depending on the lookup values we get a policy is generated.

Above tree editor window has two parts:

**Tree (upper section) - represent hierarchical structure of nodes.**

A node can have elements and sub nodes. By default tree is rendered as per policy builder configuration for DM tree.

- Node can be added copied and deleted.
  - To add more nodes right-click on the node and select Add Child. This will open a popup for child selection. Select the child and click Ok to add node in the tree.
  - To copy node right-click on the particular node and select Copy. This will create copy of the node. All children and the elements with values under the node will get copied to the new node. Element values can then be edited as required.
• To delete node right-click on the particular node and select Delete. This will delete the node.

• Node can have valid values pulled from custom reference data table. This can be configured in Policy builder.

• If the node is configured to pull valid values from the custom reference data table in Policy builder text field with Select button will be displayed against element in Control Center. When you click on Select a pop-up will be displayed for custom reference data selection. Pop-up will look like as below.

All elements of the node will be read only. Name element will have read only text field with Select and Clear button. Currently there is no support for data to be edited or deleted from the custom reference data table.

*Figure 34: Select Name Dialog Box*

- Validation of the tree is represented by the color of the nodes
  - Green color represents no error.
  - Red color represents error in elements.
  - Single folder represent node can't be repeated.
  - Multi folder represent node can be repeated.

**Elements (lower section) - represent leaf nodes of the tree that contain actual value.**

Element can have value of type integer text binary or boolean. For each type of value there is a specific editor:

- integer - text field with only integer allowed
- binary - text field with only 0 1 allowed
- text - text field with alphanumeric allowed
• boolean - check box

Element can have valid values pulled from list of valid values or values pulled from custom reference data table. CRD values can be configured in Policy builder.

If the element is configured to pull valid values from the list of values in Policy builder a drop-down list will be displayed against the element in Control center.

Data cannot be edited or deleted in Policy builder once published. New valid values can still be added but existing cannot be edited or deleted after published.

Figure 35: Parameter Selection

If the element is configured to pull valid values from the custom reference data table in Policy builder text field with Select button will be displayed against element in Control Center. When you click on Select a pop-up will be displayed for custom reference data selection. On selecting data get populated for the element.

Currently there is no support for data to be edited or deleted from the custom reference data table.
There are certain validations provided to prevent invalid trees get saved in database.

- Tree name cannot be empty.
- All mandatory nodes must be present.
- All mandatory elements of the node must have value. All mandatory elements are represented by bold letters.
- Tree deletion is not allowed if tree is attached to Group.
- Each node have upper limit. If upper limit is reached no more nodes can be added. Upper limit equals to 0 means unlimited.

### Add New DM Tree

Perform the following steps to add a new DM tree:

**Step 1**  
In the Control Center Interface, navigate to **DM Configuration**.

**Step 2**  
On the left **DM Trees** panel, select a DM tree.

**Step 3**  
Click **add**.  
**DM Trees** dialog box is displayed.

**Step 4**  
Enter a name in the **DM Tree Name** field.

**Step 5**  
Select a node in the tree.  
Data dialog box is displayed where you can populate data.
Step 6  Enter the values in each field as described below:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the policy tree.</td>
</tr>
<tr>
<td>RulePriority</td>
<td>Priority of the rule being sent.</td>
</tr>
<tr>
<td>Roaming</td>
<td>Select to enable the policy if the UE is in roaming.</td>
</tr>
<tr>
<td>PLMN</td>
<td>Public Land Mobile Network (PLMN) code that the policy serves.</td>
</tr>
<tr>
<td>UpdatePolicy</td>
<td>When enabled, updates the existing policy.</td>
</tr>
</tbody>
</table>

Step 7  Click Save.

Copy a DM Tree Node

Perform the following steps to copy a DM Tree node:

Step 1  Select a created node and right click.

For information on node creation, refer to Add New DM Tree topic.

Step 2  Select copy.

The newly created node is auto selected.

Note

1. If the name copied node is configured then "_copy" will be appended to the name of the copied node.
2. If the name copied node is not configured then "_copy" will not be appended to the name of the copied node.
3. Only the name of root node of the copied node will be appended with "_copy" and not the names of child nodes of the copied node.

Create Dynamic MO Tree Nodes with Values from CRD Table

Previously, in Control Center, when creating an instance of an MO Tree, you had to manually enter each node when you create multiple nodes that are of the same type but different values.

To create a MO Tree nodes that dynamically takes values from a CRD table, you must add a group key column in the node’s CRD Table and configure a node in the DM Tree schema with the value referencing the group key column from the node’s schema in the CRD.

Then ANDSF Server can serve an MO Tree with all the nodes that have a specific group key configured in Control Center.

In Policy Builder, assign a group key column to each of the nodes as described:

1. In Reference Data > Custom Reference Data Table, configure a group key column (example: Access_Priority) in the CRD Table.
2. In **Reference Data > DM Configuration**, configure a node (example: Access_Priority) in the DM Tree schema referencing the CRD Table.

   Ensure you set the **Valid Values** to Valid Values pulled from Subscriber CRD Table with Node Relation **One to Many**.

3. Create an element (example: Access_Key) in the node (example:Access_Priority) where the name of the element is defined in group key column.

   Ensure that the key is not bound to any values.
In Control Center, assign a Group to all the nodes:

1. In **Configuration > Reference Data**, open the node's CRD table and add all the valid entries of the node to a particular group with a group key. Assign a suitable name for the group (example: silver).

2. Again in **Configuration > Reference Data**, edit the node and enter a Name (CRD table name) and Lookup key value.
DM Tree Groups

In DM Tree Groups we can group one or more DM Trees together so that lookup can be done on multiple trees. DM Tree grouping can be done by clicking on available groups on the left side panel under DM Tree Grouping. It would open a window (as shown below) showing list of groups already created.

The above window also contains an add button clicking on which would open a tab in the same window (as shown below).
This new tab would contain the following fields:

- **Name:** This is used for giving a unique name to the group. This field cannot be left blank. Also if name is not unique creation of it would throw an error.
- **DM Tree Group:** This has two boxes left box lists all the available DM Trees and right box is the one which contains selected DM Trees. In between these two boxes there are arrow buttons given to move tree(s) from left to right and vice versa. All the trees put under right box would become part of the group. Right box cannot be left empty.
- Once tree(s) are selected click on Save to create the group.
- In order to reset the entire view of the group creation click **Reset.** Reset would take you to initial view of the group creation as if you have click on the add button on the first window above.

**DM Tree Lookups**

In DM Tree Lookups we define a lookup name and value and associated DMTreeGroup Name. This value corresponds to the Lookup Structure we created in Policy Builder. For example we define geo_location lookup entries assigning corresponding DMT Group name. This lookup is based on a certain DMT Lookup (in this case Policy_EXT_GEO_LOC_STATIC) in Policy Builder. Example configurations are shown below:
DM Tree Lookups - Apple Devices

**Figure 43: DM Tree Lookups - Policy EXT GEO LOC STATIC**

In \textit{Configuration} tab the corresponding lookup (GEO\_Location) fields must be mapped to certain values.

**Figure 44: Geo Location**

DM Tree Lookups - Android

Here we define TGPP\_Location and WLAN\_Location lookup entries assigning corresponding DMT Group name. This lookup is based on a certain DMT Lookup (in this case Policy\_EXT\_TIER\_CISCO (We can use default lookups too)) in Policy Builder.
In **Configuration** Tab, the corresponding lookup (**TGPP_Location** and **WLAN_Location**) fields must be mapped to certain values.

**Figure 46: TGPP Location**

<table>
<thead>
<tr>
<th>Location Name</th>
<th>PLMN Code</th>
<th>Tracking Area Code</th>
<th>Location Area Code</th>
<th>UMTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>LocationName1</td>
<td>PLMN1</td>
<td>TA1</td>
<td>LA1</td>
<td></td>
</tr>
<tr>
<td>LocationName2</td>
<td>PLMN2</td>
<td>TA2</td>
<td>LA2</td>
<td></td>
</tr>
<tr>
<td>LocationName3</td>
<td>PLMN3</td>
<td>TA3</td>
<td>LA3</td>
<td></td>
</tr>
<tr>
<td>LocationName4</td>
<td>PLMN4</td>
<td>TA4</td>
<td>LA4</td>
<td></td>
</tr>
<tr>
<td>LocationName5</td>
<td>PLMN5</td>
<td>TA5</td>
<td>LA5</td>
<td></td>
</tr>
<tr>
<td>LocationName6</td>
<td>PLMN6</td>
<td>TA6</td>
<td>LA6</td>
<td></td>
</tr>
<tr>
<td>LocationName7</td>
<td>PLMN7</td>
<td>TA7</td>
<td>LA7</td>
<td></td>
</tr>
<tr>
<td>LocationName8</td>
<td>PLMN8</td>
<td>TA8</td>
<td>LA8</td>
<td></td>
</tr>
<tr>
<td>LocationName9</td>
<td>PLMN9</td>
<td>TA9</td>
<td>LA9</td>
<td></td>
</tr>
<tr>
<td>LocationName10</td>
<td>PLMN10</td>
<td>TA10</td>
<td>LA10</td>
<td></td>
</tr>
</tbody>
</table>
iPhone Mobile Configuration

ANDSF provides user interface for creating iphone mobile configuration which can be installed on iphone device. In this release, a graphical user interface in control center component has been added which is part of ANDSF server and thereby enabling user to create and manage iphone mobile configuration. This configuration internally creates a self signed certificate implicitly for geo-location based services to work properly:

**Step 1** Login to the Control Center.

**Step 2** Select **DM Configuration** tab and click **Mobile Config (iOS device)** from the left pane to open **View Mobile Config**.

**Step 3** Click **add** on the right side to open **New Mobile Config** pane.

**Step 4** Click **General** from left pane to open the interface as shown:
Under **General**, you can enter the following parameters based on your requirements:

**Table 18: Mobile Config Parameters - General**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the profile. This name is displayed to the user during its installation on iphone device.</td>
</tr>
<tr>
<td>Identifier</td>
<td>A reverse-DNS style identifier (com.example.myprofile for example) that identifies the profile. This string is used to determine whether a new profile should replace an existing one or should be added.</td>
</tr>
<tr>
<td>Organization</td>
<td>A human-readable string containing the name of the organization that provided the profile.</td>
</tr>
<tr>
<td>Description</td>
<td>Brief message that will be displayed during profile installation.</td>
</tr>
<tr>
<td>Consent Message</td>
<td>Description of the profile shown on the Detail screen for the profile. This should be descriptive enough to help the user decide whether to install the profile.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Profile removal allowed</td>
<td>Indicates if profile removal is:</td>
</tr>
<tr>
<td></td>
<td>• Never allowed</td>
</tr>
<tr>
<td></td>
<td>• Always allowed</td>
</tr>
<tr>
<td></td>
<td>• Allowed with Password</td>
</tr>
<tr>
<td>Profile Expiry</td>
<td>Indicates if profile auto removal:</td>
</tr>
<tr>
<td></td>
<td>• Is never allowed</td>
</tr>
<tr>
<td></td>
<td>• Should happen on given date</td>
</tr>
<tr>
<td></td>
<td>• Should happened after given number of days</td>
</tr>
</tbody>
</table>

**Step 5**  After configuring the required parameter, click Save to save the general configuration.

**Step 6**  Click **Wi-Fi** from left to open an interface as shown:

*Figure 49: New Mobile Config - Wi-Fi*

You can either use **Search** to search for already added SSID or can add a new SSID by clicking add.

**Step 7**  To add a new SSID, click **add** to open an interface as shown:
You can enter the following parameters based on your requirements:

**Table 19: New Mobile Config - New SSID Parameters**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ssid</td>
<td>Service set identifier.</td>
</tr>
<tr>
<td>Auto Join</td>
<td>Decision to automatically join the target network.</td>
</tr>
<tr>
<td>Hidden Network</td>
<td>Decision to apply this profile to network even if it is not open or broadcasting.</td>
</tr>
<tr>
<td>Security Type</td>
<td>Security setting used during network connection. Different security types are as follows</td>
</tr>
<tr>
<td></td>
<td>• None No security.</td>
</tr>
<tr>
<td></td>
<td>• WEP</td>
</tr>
<tr>
<td></td>
<td>• WPA/WPA2</td>
</tr>
<tr>
<td></td>
<td>• Any (Personal)</td>
</tr>
<tr>
<td></td>
<td>• WEP Enterprise</td>
</tr>
<tr>
<td></td>
<td>• WPA/WPA2 Enterprise</td>
</tr>
<tr>
<td></td>
<td>• Any (Enterprise)</td>
</tr>
</tbody>
</table>

* Based on security type selected above we get different parameter options
• For WEP we need to provide a password in **Password** field for simple authentication.
• For WPA/WPA2 we need to provide a password in **Password** field for simple authentication.
• For Any (Personal) we need to provide a password in **Password** field for simple authentication.

For **WEP Enterprise WPA/WPA2 Enterprise, Any (Enterprise)**, we need to select Protocols and provide **Authentication** parameters.

*Figure 51: New Mobile Config - Protocols*

You can enter the following parameters based on your requirements:

*Table 20: New Mobile Config - Protocol Parameters*

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Protocols</strong></td>
<td></td>
</tr>
<tr>
<td>Accepted EAP Types</td>
<td>TLS TTLS LEAP PEAP EAP-FAST EAP-SIM</td>
</tr>
<tr>
<td>EAST-FAST</td>
<td>Use PAC (Protected Access Credentials) Provision PAC Provision PAC anonymously.</td>
</tr>
<tr>
<td><strong>Authentication</strong></td>
<td></td>
</tr>
<tr>
<td>Username</td>
<td>Username for enterprise authentication.</td>
</tr>
<tr>
<td>Use Per-Connection Password</td>
<td>Request password during connection and send with authentication.</td>
</tr>
<tr>
<td>Parameters</td>
<td>Type</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>Password</td>
<td>Password for enterprise authentication.</td>
</tr>
<tr>
<td>Outer Identity</td>
<td>Externally visible identification (for TTLS PEAP and EAP-FAST).</td>
</tr>
</tbody>
</table>

If TTLS is selected as Accepted EAP Types protocol then we need to select Inner Identity from the drop-down list as shown:

*Figure 52: New Mobile Config - Protocols*

---

**Step 8**

After configuring all the required parameters in Wi-Fi pane, click **Save** to save the wi-fi configuration.

---

**iOS Mobile Configuration**

ANDSF supports capability to dynamically generate mobile-config file and sign it with self-generated certificate or provisioned certificate as well providing a way to upload static mobile-config files generated outside the ANDSF system. Also ANDSF hosts the mobile-config and certificates files accessible over HTTPS interface.

To configure iOS Mobile Configuration we need to configure following components of ANDSF system:

- **Control Center**: We need to configure and create mobile-config and certificate file.
- **ANDSF**: This component is responsible for interacting with database and file system to store the mobile-config file created by Control Center component and also storage of certificate file. It would also provide interface for CRUD operation. This component is also responsible for including mobile-config and certificate URL in the ANDSF policy which are responded to the client device.
- **Policy Builder**: This component provides graphical interface to provision directory location for mobile-config and certificate files. This also enable user to add custom response headers while responding mobile-config and certificate over https.

---

**Step 1**

**Mobile Configuration File Creation**
Login to the Control Center and configure Mobile Configuration file according to the steps mentioned in iPhone Mobile Configuration.

Step 2  Certificate Generation

In Control Center, click Certificates under Mobile Config (iOS device) to open Certificates window. You can either use Search to search for already added certificates or can add a new Certificate by clicking add.

Step 3  To add a certificate, click add to open Certificate interface as shown:

Step 4  Click Browse to select the required certificate from your local machine.

Step 5  Click Upload to upload the certificate to the Control Center.

Step 6  Serializing and Signing the Mobile Configuration File

Click Trusted CA under Mobile Config (iOS device) to open Trusted CAs window. You can either use Search to search for already added certificates or can add a new certificate by clicking add.

Trusted-CA is used to digitally sign the entire mobile configuration file.

Step 7  To add a certificate, click add to open Certificate interface as shown:
Step 8  Click **Browse** to select the required certificate as shown above. Enter the required **Password** and **Pvt Key Password** in the respective fields.

Step 9  After entering the required information, click **Upload** to uploaded the trusted certificate to the database.

Step 10 Click **Mobile Config (iOS device)** from left pane to open mobile configuration window.

Step 11 Search for the mobile configuration file for which you want to assign the certificate.

Step 12 Click the mobile configuration file to open Edit interface as shown:
Step 13 Click Trusted-CA check box. Trusted-CA is used to digitally sign the entire mobile configuration file.

Step 14 Click Select Trusted-CA to open an interface from where you can select the Trusted CA. Once selected the CA name appears near Select Trusted-CA box.

Step 15 Select the certificate expiry date from Cert Expiry calendar box.

Step 16 Click Generate Cert to embed an internally generated self signed certificate. The generated certificate appears in the Certificate pane.

Step 17 After configuring all the parameters, click Update to assign a certificate to an existing iOS Mobile Configuration file.

Step 18 Storage

Login to the Policy Builder.

Step 19 Click Reference Data tab and select Systems from left side.

Step 20 Under Summary, select Andsf Configuration to open the pane as shown:
Step 21 Configure the required parameters. For more information on the parameters, refer to ANDSF Plugin Configuration.

Using REST APIs

You can use REST APIs to perform DM Tree Lookup, DM Tree Group, and DM Tree operations. You can Create, Get, Update, and Delete DM Tree Lookups, DM Tree Groups, and DM Trees based on their "name" index.

DM Tree Lookup

You can use the insert, get, update, and delete APIs for managing DM Tree Lookups.

Insert DM Tree Lookup to ANDSF Database

API path:
/andsf/api/lookup/<lookuptype>

**API HTTP Method:**

POST

**API Description:**

Insert the ANDSF DM Tree lookup record into the ANDSF database. The request body must contain the JSON of the DM Tree lookup record to be inserted. The request body must be formatted as per the following schema:

```json
{
    "name": "<new lookup name>",
    "lookupKeyMap": {
        "WLAN_Location": "<name of wlan location>",
        "TIER": "<tier of subscriber>",
        "TGPP_Location": "<name of tgpp location>"
    },
    "groupName": "<group to be assigned to the lookup>
}
```

**name field:** {Mandatory in request}

The value of the name field can be given in the request body by the client. The client has to ensure that the name field is unique for each lookup to be inserted, else the API returns an Error Response.

**HTTP Status Codes in Response:**

1. 200 (CREATED): This status code is sent by the ANDSF API endpoint when the DM Tree Lookup is inserted successfully in the ANDSF database.

2. 400 (BAD REQUEST): This status code is sent by the ANDSF API endpoint when the body of the request is missing some elements or have invalid elements.

3. 500 (INTERNAL SERVER ERROR): This status code is returned if there are any exceptions while processing the request (for example, MongoDB going down).

---

**Get DM Tree Lookup Record By Name**

**API path:**

/andsf/api/lookup/<lookuptype>/<lookup name>

**API HTTP Method:**

GET

**API Description:**

Retrieves the lookup record from the ANDSF database by using the DM Tree Lookup type and the lookup name to query. If the lookup does not exist in the database, the response contains an error code. The response contains a JSON document of the lookup if the lookup type and name exists in the database.

An example of a response:

```json
{
    "name": "newLookup",
    "groupName": "NewGroup",
    "lookupKeyMap": {
        "WLAN_Location": "UseCase_Tier_WLAN",
        "TIER": "Gold",
        "TGPP_Location": "UseCase"
    }
}
```
Lookup type field: {Mandatory in request path}
The Lookup Type field must have the collection or the lookup types that need to be searched in the ANDSF database. If the lookup type is not present, the response gives an error code.

Lookup name field: {Mandatory in request path}
The Lookup name field must have the name of the lookup that needs to be searched in the ANDSF database. If the lookup name is not present, the response gives an error code.

HTTP Status Codes in Response:
1. 200 (OK): This status code is sent by the ANDSF API endpoint when the DM Tree Lookup record is successfully retrieved in the ANDSF database.
2. 404 (NOT FOUND): This status code is sent by the ANDSF API endpoint if the DM Tree Lookup does not exist in the ANDSF database.
3. 500 (INTERNAL SERVER ERROR): This status code is returned if any exception occurs when processing the request. For example, if MongoDB goes down.

Update DM Tree Lookup Record in Database by Querying Name

API path:
/andsf/api/lookup/<lookuptype>

API HTTP Method:
PUT

API Description:
Update the ANDSF lookup record from the ANDSF database by using the lookuptype field to query the database. The request body must contain the JSON of the lookup record to be updated. The request body must be formatted as per the following schema:

```
{
  "name": "crudLookupNew1",
  "groupName": "<group name>",
  "lookupKeyMap": {
    "WLAN_Location": "<wlan location>",
    "TIER": "<tier>",
    "TGPP_Location": "<tgpp location>"
  }
}
```

name field: {Mandatory in request}
The value of the name field can be given in the request body by the client. The client has to ensure that the name field is unique for each DM Tree lookup to be inserted, else the API returns an Error Response.

HTTP Status Codes in Response:
1. 200 (NO CONTENT): This status code is sent by the ANDSF API endpoint when the DM Tree lookup record is successfully updated into the ANDSF database.
2. 400 (BAD REQUEST): This status code is sent by the ANDSF API endpoint when the body of the request is missing some elements or have invalid elements.
3. 404 (NOT FOUND): This status code is sent by the ANDSF API endpoint when the DM Tree Lookup does not exist in the ANDSF database.

4. 500 (INTERNAL SERVER ERROR): This status code is returned if there are any exceptions while processing the request (for example, MongoDB going down).

Delete DM Tree Lookup Record in Database by Querying Lookup Name

**API path:**
/andsf/api/lookup/<lookuptype>/<lookupname>

**API HTTP Method:**
DELETE

**API Description:**
Delete the ANDSF lookup record from the ANDSF database by using the lookup name field in the query. If the lookup name is not present in the database, the API returns an error code.

**HTTP Status Codes in Response:**
1. 204 (NO CONTENT): This status code is sent by the ANDSF API endpoint when the DM Tree lookup record is successfully deleted from the ANDSF database.
2. 404 (NOT FOUND): This status code is sent by the ANDSF API endpoint when the DM Tree Lookup does not exist in the ANDSF database.
3. 500 (INTERNAL SERVER ERROR): This status code is returned if any exception occurs when processing the request. For example, if MongoDB goes down.

**DM Tree Group**

You can use the insert, get, update, and delete APIs for managing DM Tree Groups.

**Insert a DM Tree Group in ANDSF Database**

**API path:**
/andsf/api/group/<DM Tree Group Template type>

**API HTTP Method:**
POST

**API Description:**
Inserts an ANDSF DM Tree Group record into the ANDSF database. The request body must contain the JSON of the DM Tree Group record to be inserted. The request body must be formatted as described in the following schema:

```json
{
    "name": "ANDSF_CRUD_Test_grp",
    "dmtNames": [
        "UseCase_SSID", "UseCase_SSID_Tier"
    ]
}
```
name field: {Mandatory in request}

The value of the name field must be included in the request body from the client. The client must ensure that the name field is unique for a group to be inserted; otherwise, the API returns an Error Response.

dmtNames field: {Mandatory in request}

The value of the dmtNames field must be included in the request body from the client. dmtNames must be given in the form of a list.

HTTP Status Codes in Response:

1. 200 (CREATED): This status code is sent by the ANDSF API endpoint when the DM Tree Group record is inserted successfully in the ANDSF database.
2. 400 (BAD REQUEST): This status code is sent by the ANDSF API endpoint when the body of the request is missing some elements or has invalid elements.
3. 500 (INTERNAL SERVER ERROR): This status code is returned if there are any exceptions while processing the request (for example, MongoDB going down).

Get DM Tree Group By Name

API path:
/andsf/api/group/<DM Tree Group Template Type>/<DM Tree Group Name>

API HTTP Method:
GET

API Description:
Retrieves DM Tree Group record from the ANDSF database by using the DM Tree Group Template Type and the DM Tree Group Name to query. If the DM Tree Group does not exist in database, the response contains an error code. The response contains the JSON document of the DM Tree Group if the DM Tree Group Type and DM Tree Group Name exists in the database.

An example of a response:
{
   "name": "ANDSF_CRUD_Test_grp",
   "dmtNames": [
       "UseCase_SSID",
       "UseCase_SSID_Tier"
   ]
}

DM Tree Group Template Type field: {Mandatory in request path}

The DM Tree Group Template Type field must have the collection that needs to be searched in the ANDSF database (for example, ISMP, ISMP_EXT, etc.). If the DM Tree Group Template Type is not present, the response gives an error code.

DM Tree Group Name field: {Mandatory in request path}

The DM Tree Group name field must have the name of the DM Tree Group record to be searched in the ANDSF database. If the DM Tree Group name is not present, the response gives an error code.

HTTP Status Codes in Response:
1. 200 (OK): This status code is sent by the ANDSF API endpoint when the DM Tree Group is successfully retrieved in the ANDSF database.

2. 404 (NOT FOUND): This status code is sent by the ANDSF API endpoint if the DM Tree Group does not exist in the ANDSF database.

3. 500 (INTERNAL SERVER ERROR): This status code is returned if any exception occurs when processing the request. For example, if MongoDB goes down.

Update DM Tree Group by Name

**API path:**

/andsf/api/group/<DM Tree Group Template Type>

**API HTTP Method:**

PUT

**API Description:**

Update ANDSF DM Tree Group record in the ANDSF database by using the DM Tree Group Template Type field to query. The request body must contain the JSON of the DM Tree Group record to be updated. The request body must be formatted as described in the following schema:

```
{
  "name": "ANDSF_CRUD_Test_grp",
  "dmtNames": [
    "UseCase_PGF"
  ]
}
```

**name field:** {Mandatory in request}

The value of the name field can be given in the request body by the client. The client has to ensure that the name field is unique for a group to be inserted, else the API returns an Error Response.

**dmtNames field:** {Mandatory in request}

The value of the dmtNames field must be included in the request body from the client. dmtNames must be given in the form of a list.

**HTTP Status Codes in Response:**

1. 200 (NO CONTENT): This status code is sent by the ANDSF API endpoint when the DM Tree Group record is successfully updated into the ANDSF database.

2. 400 (BAD REQUEST): This status code is sent by the ANDSF API endpoint when the body of the request is missing some elements or have invalid elements.

3. 404 (NOT FOUND): This status code is sent by the ANDSF API endpoint when the DM Tree Group does not exist in the ANDSF database.

4. 500 (INTERNAL SERVER ERROR): This status code is returned if there are any exceptions while processing the request (for example, MongoDB going down).

Delete DM Tree Group by Name

**API path:**
API HTTP Method:
DELETE

API Description:
Delete the DM Tree Group record from the ANDSF database by using the DM Tree Group name field in the query. If the DM Tree Group is not present in the database, the API returns an error code.

HTTP Status Codes in Response:
1. 204 (NO CONTENT): This status code is sent by the ANDSF API endpoint when the DM Tree Group record is successfully deleted from the ANDSF database.
2. 404 (NOT FOUND): This status code is sent by the ANDSF API endpoint when the DM Tree Group does not exist in the ANDSF database.
3. 500 (INTERNAL SERVER ERROR): This status code is returned if any exception occurs when processing the request. For example, if MongoDB goes down.

DM Tree

You can use the insert, get, update, and delete APIs for managing DM Trees.

Insert a DM Tree to ANDSF Database

API path:
/andsf/api/tree/<DM Tree template type>

API HTTP Method:
POST

API Description:
Insert an ANDSF DM Tree record into the ANDSF database. The request body must contain the JSON of the DM Tree record to be inserted. The request body must be formatted as described in the following schema:

```json
{
    "priority": null,
    "name": "UseCase_Hello",
    "templateName": "ISMP_EXT",
    "rootNode": {
        "name": "UseCase_Hello",
        "value": null,
        "type": null,
        "nodes": [
            {
                "name": "Policy",
                "value": null,
                "type": null,
                "nodes": [
                    {
                        "name": "UseCase_Hello",
                        "value": null,
                        "type": null,
                        "nodes": [
                        ]
                    }
                ]
            }
        ]
    }
}
```
```
{
    "name": "RulePriority",
    "value": 1,
    "type": "Integer",
    "nodes": [],
    "xnode": false
},
{
    "name": "Roaming",
    "value": false,
    "type": "Boolean",
    "nodes": [],
    "xnode": false
},
{
    "name": "PLMN",
    "value": "template_PLMN",
    "type": "Text",
    "nodes": [],
    "xnode": false
},
{
    "name": "UpdatePolicy",
    "value": true,
    "type": "Boolean",
    "nodes": [],
    "xnode": false
},
{
    "name": "Ext",
    "value": null,
    "type": null,
    "nodes": [
        {
            "name": "SSIDTypeWLAN_Location2",
            "value": null,
            "type": null,
            "nodes": [
                {
                    "name": "SSID_Table_values",
                    "value": null,
                    "type": null,
                    "nodes": [
                        {
                            "name": "SSID_KEY",
                            "value": "*",
                            "type": "Text",
                            "nodes": [],
                            "xnode": false
                        }
                    ],
                    "xnode": false
                }
            ],
            "xnode": true
        }
    ],
    "xnode": false
},
{
    "name": "PrioritizedAccess",
    "value": null,
    "type": null,
    "nodes": []
}
```
"name": "Employee_access",
"value": null,
"type": null,
"nodes": [
    {
        "name": "AccessTechnology",
        "value": "3",
        "type": "Integer",
        "nodes": [],
        "xnode": false
    },
    {
        "name": "AccessId",
        "value": "employee",
        "type": "Text",
        "nodes": [],
        "xnode": false
    },
    {
        "name": "AccessNetworkPriority",
        "value": 1,
        "type": "Integer",
        "nodes": [],
        "xnode": false
    }
],
"xnode": true
},
{
    "name": "Guest_access",
    "value": null,
    "type": null,
    "nodes": [
        {
            "name": "AccessTechnology",
            "value": "3",
            "type": "Integer",
            "nodes": [],
            "xnode": false
        },
        {
            "name": "AccessId",
            "value": "guest",
            "type": "Text",
            "nodes": [],
            "xnode": false
        },
        {
            "name": "AccessNetworkPriority",
            "value": 2,
            "type": "Integer",
            "nodes": [],
            "xnode": false
        }
    ],
    "xnode": true
},
"xnode": false
],
"xnode": false
DM Tree structure in the request body has to match the schema for the specific DM Tree type in Policy Builder.

**Fields of the JSON Tree**

- **name**: Name of the DM Tree
- **templateName**: Name of the template of the DM Tree (the name of the particular schema of the DM Tree in Policy Builder)
- **priority**: Null by default
- **rootNode**: The tree structure

**Fields of the rootNode Tree**

- **name**: Name of the DM Tree
- **value**: Null by default
- **type**: Null by default
- **nodes**: Policy node

**Fields of other nodes and elements**

- **name**: Name of the node (for example, TGPP Location, Policy, SSIDTypeWLAN_Location2, any of the nodes or elements that are defined in the schema in Policy Builder)
- **value**: If the node represents an actual node in the DM Tree, this is null. If it represents an element, the value is the value of that element.
- **type**: If the node represents an actual node in the DM Tree, this is null. If it represents an element, the type is the type of that element.
- **nodes**: If the node represents an actual node in the DM Tree, this has all the nodes/elements under this particular node. If it represents an element, this is an empty list.

**HTTP Status Codes in Response:**

1. **200 (CREATED)**: This status code is sent by the ANDSF API endpoint when the DM Tree record is inserted successfully in the ANDSF database.
2. **400 (BAD REQUEST)**: This status code is sent by the ANDSF API endpoint when the body of the request is missing some elements or has invalid elements.
3. **500 (INTERNAL SERVER ERROR)**: This status code is returned if any exception occurs when processing the request. For example, if MongoDB goes down. Also, if the JSON body in the request is not as per the schema, 500 error code is returned.

**Get DM Tree By Name**

**API path:**
Update DM Tree by Name

API path:
/andsf/api/tree/<DM Tree Template Type>

API HTTP Method:
PUT

API Description:
Update the ANDSF DM Tree record in the ANDSF database by using the DM Tree Template Type field to query the database. The request body must contain the JSON of the DM Tree record to be updated. The request body must be formatted as described in Get DM Tree By Name, on page 72.

HTTP Status Codes in Response:
1. 200 (NO CONTENT): This status code is sent by the ANDSF API endpoint when the DM Tree record is successfully updated into the ANDSF database.
2. 400 (BAD REQUEST): This status code is sent by the ANDSF API endpoint when the body of the request is missing some elements or have invalid elements.
3. 404 (NOT FOUND): This status code is sent by the ANDSF API endpoint when the DM Tree does not exist in the ANDSF database.
4. 500 (INTERNAL SERVER ERROR): This status code is returned if any exception occurs when processing the request. For example, if MongoDB goes down. Also, if the JSON body in the request is not as per the schema, 500 error code is returned.

Delete DM Tree by Name

API path:
/andsf/api/tree/<DM Tree Template Type>/<DM Tree Name>
API HTTP Method:
DELETE

API Description:
Delete the DM Tree from the ANDSF database by using the DM Tree name field in the query. If the DM Tree name is not present in the database, the API returns an error code.

HTTP Status Codes in Response:
1. 204 (NO CONTENT): This status code is sent by the ANDSF API endpoint when the DM Tree is successfully deleted from the ANDSF database.
2. 404 (NOT FOUND): This status code is sent by the ANDSF API endpoint when the DM Tree does not exist in the ANDSF database.
3. 500 (INTERNAL SERVER ERROR): This status code is returned if any exception occurs when processing the request. For example, if MongoDB goes down.
Custom Reference Data Tables

Read-Only templates have been created in Policy Builder under Custom Reference Data Tables based on the 3GPP TS 24.312 specification. The user can populate the data for these master tables in Control Center.

The following individual Read-Only templates have been created under Custom Reference Data Tables in Policy Builder.

In Control Center make sure that all the above tables are created with proper schema. The table values will be populated based on operator requirements.

Read-only Template for ANDSF Client

*Figure 57: Read-only Template for ANDSF Client*
### Read-only Template for WLAN Location

**Figure 58: Read-only Template for WLAN Location**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Key</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Location Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default</td>
<td>Location Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Priority</td>
<td>Location Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameters</td>
<td>Parameters</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Values</td>
<td>Values</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Details</td>
<td>Details</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Actions</td>
<td>Actions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Copy</td>
<td>Current Custom Reference Data Table</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Read-only Template for TGPP Location

**Figure 59: Read-only Template for TGPP Location**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Key</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default</td>
<td>Default</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameters</td>
<td>Parameters</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Values</td>
<td>Values</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Details</td>
<td>Details</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Actions</td>
<td>Actions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Copy</td>
<td>Current Custom Reference Data Table</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Read-only Template for Geo Location

## Figure 60: Read-only Template for Geo Location

<table>
<thead>
<tr>
<th>Column</th>
<th>Title</th>
<th>Type</th>
<th>Description</th>
<th>Validation</th>
<th>Key</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>LocationName</td>
<td>String</td>
<td>Location Name</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Str</td>
<td>LocationID</td>
<td>String</td>
<td>Location ID</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Id</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Columns Details**

**Valid Values**

- The values allowed in the LocationName column
- A list of valid values

**Validation**

- Validation used by the database
- Regular Expression

**Row-based Display**

- Which rows are shown when a message is retrieved
- Matching Operation

**Subgrid**

- Subgrid Reference Table
- Subgrid Result column
- Subgrid Subgrid Reference Table
- Subgrid Subgrid Result column

**List of Custom Reference Data Tables**

- List of custom reference data tables

Control Center Configurations

- Subscribers, on page 79

Subscribers

This section provides description and examples to find, create or edit subscriber information. For more information, refer to CPS CCI Guide for Full Privilege Administrators for this release.

Find a Subscriber

After you log in to the Control Center, the default page of the Cisco Control Center interface displays the Subscribers > Find Subscriber pane as shown below:

*Figure 61: Cisco Control Center Interface*

The Subscribers tab lets you look at details for a subscriber, any subaccounts under a subscriber, and any sessions a subscriber may have.

Finding a subscriber is often the first step in performing other tasks for a subscriber. To list all subscribers, leave the Credential or Name field blank. The maximum number of records returned is 1000.
Create a Subscriber

To import a large number of subscribers, use your own subscriber profile repository (SPR) software.

To create a single subscriber, perform the following steps:

Step 1  Click **Subscribers > Subscribers node > Create Subscriber** to open window on the right as shown below:
Create Subscriber

a) Type in a Credential that specifically identifies the new subscriber.

   A credential is a unique identifier that subscribers and subaccount customers use to log in. Typically, a credential is defined when you create the subscriber or subaccount. The credential can be any letter or number with a length of one or more alphanumeric characters. Best practice is to not use special characters, although apostrophe and hyphen may be part of a name.

b) Enter the subscriber’s name, first and last, in the Name field.

c) Click Save to create the subscriber with name, and credential defined. You can return later and fill in more details later.

Step 2  Click Save & Continue to provide more details for the new subscriber at this time.

Save Subscriber

General

The following parameters can be configured in the General window:
### General Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>This single field is for both the first and last name of the subscriber.</td>
</tr>
<tr>
<td>Domain</td>
<td>This drop-down list lets you assign the subscriber a domain. Domains themselves are created in the Cisco Policy Builder interface.</td>
</tr>
<tr>
<td>Status</td>
<td>The Status field shows these four states for a subscriber:</td>
</tr>
<tr>
<td></td>
<td>- Active: Able to use services.</td>
</tr>
<tr>
<td></td>
<td>- Deleted: No longer part of your system.</td>
</tr>
<tr>
<td></td>
<td>- Suspended: In the system, but not able to use services.</td>
</tr>
<tr>
<td></td>
<td>- Inactive: If the subscriber is not active and has no balance, no services for a given period of time, then that subscriber is moved to Inactive state.</td>
</tr>
<tr>
<td></td>
<td>By default, status for a new subscriber is always ACTIVE.</td>
</tr>
<tr>
<td>Start Date and</td>
<td>Use the calendar widget to specify the start and stop date and time of service to the subscriber.</td>
</tr>
<tr>
<td>End Date</td>
<td></td>
</tr>
<tr>
<td>Role</td>
<td>When the subscriber logs in to your subscriber portal, this field determines how much read-write privilege is granted to them.</td>
</tr>
<tr>
<td></td>
<td>The primary subscriber is always granted the Write All role, permitting editing of all subaccounts under it. For example, the primary subscriber might be a team leader responsible for team member accounts.</td>
</tr>
<tr>
<td></td>
<td>If a user is not a primary subscriber, they are created as a subaccount user under a primary subscriber. Assign roles for subaccount users in the Details &gt; Subaccounts window.</td>
</tr>
<tr>
<td></td>
<td>Be sure to assign the access privileges of a subaccount user. When first created, they have Write All privileges.</td>
</tr>
<tr>
<td></td>
<td>- Write All: Can read and write all information for themselves and any subaccounts.</td>
</tr>
<tr>
<td></td>
<td>- Read All: Can view all portal information.</td>
</tr>
<tr>
<td></td>
<td>- Write Self: Can read and write only their own information.</td>
</tr>
<tr>
<td></td>
<td>- Read Self: Can view only their own information.</td>
</tr>
<tr>
<td>External Id</td>
<td>Occasionally, a subscriber may need to connect with or relate to an external third-party system. This field identifies the subscriber to that external service.</td>
</tr>
<tr>
<td>Rate Plan</td>
<td>Enter predefined rate plan codes, such as prepaid or postpaid. Rate plans are created in Cisco Policy Builder and are obtained from that administrator.</td>
</tr>
<tr>
<td>Charging Id</td>
<td>A subscriber might have a unique charging ID. Using this, usage by members of a subaccount, or ‘children’ of the subscriber can be billed to their ‘parent’.</td>
</tr>
</tbody>
</table>
**Parameter** | **Description**  
--- | ---  
Authentication Type | Type of authentication: None, Basic, Digest and Simple. Currently, we support only Basic authentication.  
Username | Username for the type of authentication selected above. For None authentication type, this field will be blank.  
**Note** | Username and Name should be same.  
Password | Password for the type of authentication selected above. For None authentication type, this field will be blank.  
Custom Data | This area lets you use a look up list, or key-value pairs to keep track of other variables that concern your subscriber.  
• Code: This is the key portion of a key-value pair.  
• Value: This is the definition or value of the key-value pair.  

---

**SSID Configuration**

The SSID is a unique identifier that wireless networking devices use to establish and maintain wireless connectivity. Here we set wireless details (SSID configuration) of multiple access points for a particular subscriber.

**Step 1**  
Under the current subscriber, click **SSIDs** from the left pane to open a window on the right side as shown below:

*Figure 66: SSID*

![SSID Configuration](image)

**Step 2**  
Click **add** to open SSID List/SSID Detail window as shown below:
The following parameters can be configured under SSID List/SSID Detail:

**Table 22: SSID Configuration Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ssid Key</td>
<td>SSID name identified in SPR database.</td>
</tr>
<tr>
<td>SSIDs</td>
<td>SSID name of that access point.</td>
</tr>
<tr>
<td>Access Type</td>
<td>Access type (MAN/UNMAN).</td>
</tr>
<tr>
<td>Auth Type</td>
<td>Authentication type (WPA etc).</td>
</tr>
<tr>
<td>Username</td>
<td>Username of access point.</td>
</tr>
<tr>
<td>Password</td>
<td>Password of access point.</td>
</tr>
<tr>
<td>Login Url</td>
<td>Login URL to access point.</td>
</tr>
<tr>
<td>Config Url</td>
<td>URL of mobile-config (for AppleDevices). Format: <code>&lt;IP of QPS&gt;:8080/qps/rest/andsf/static/policy?&lt;mobile_config_name&gt;.mobileconfig</code></td>
</tr>
<tr>
<td>Verification Cert Url</td>
<td>URL of verification certificate (for AppleDevices) Format: <code>&lt;IP of QPS&gt;:8080/qps/rest/andsf/static/policy?&lt;cert_name&gt;.cer</code></td>
</tr>
</tbody>
</table>
### Parameter | Description
--- | ---
Config Message | We can set these fields for various prompt messages.
Portal Fail Message
Unmanaged Vpn Connect
Unmanaged Vpn Disconnect Prompt

An example configuration is shown below:

*Figure 68: SSID Example*

![SSID Example](image)

The configuration and certificate files should be hosted in ANDSF server for each profile.

**Custom Data (Details)**

In Custom Data we set AVP (additional value pairs) of subscriber, like tier details (gold, silver and bronze).

- Code: Tier
- Value: Gold/Silver/Bronze

**Edit a Subscriber**

**Step 1** Find the subscriber as described in Find a Subscriber, on page 79.

**Step 2** Hover over the name displayed in the Results table and click open.
Step 3  
The **Subscriber Overview** window appears.

Step 4  
From the above window, click any of the **edit** or **manage** links to change subscriber information about that topic.

Step 5  
You can also use the menu items in the menu tree on the left to display individual screens.
### Figure 71: Test Menu Items

<table>
<thead>
<tr>
<th>Subscribers</th>
<th>Test Overview</th>
</tr>
</thead>
<tbody>
<tr>
<td>Find Subscriber</td>
<td>Name: Test</td>
</tr>
<tr>
<td>Create Subscriber</td>
<td>Credentials: Test</td>
</tr>
<tr>
<td>Overview</td>
<td></td>
</tr>
<tr>
<td>Details</td>
<td></td>
</tr>
<tr>
<td>Sessions</td>
<td></td>
</tr>
<tr>
<td>Balance</td>
<td></td>
</tr>
<tr>
<td>SSIDs</td>
<td></td>
</tr>
<tr>
<td>Sessions</td>
<td></td>
</tr>
<tr>
<td>Find Subscriber Session</td>
<td></td>
</tr>
<tr>
<td>Find Network Session</td>
<td></td>
</tr>
</tbody>
</table>
CHAPTER 6

Notification Services

- Notification Services, on page 89
- Service Option Configuration, on page 96

Notification Services

Notification in Policy Builder relates to pushing messages from Policy Builder to subscribers. Service Providers can use messages to alert the subscriber to issues as well as opportunities on their network. Not only can you alert subscribers, but you can also send messages to any address you wish, perhaps system monitoring addresses.

Currently, Policy Builder offers following notification types:

- Apple Push Notifications, on page 89
- GCM Notifications, on page 92

Note

You can configure one or all notification types. By default, no notification type is configured in Policy Builder. User needs to configure the notifications based on your requirements.

For more information on how to configure Notification plug-in in Policy Builder, refer to Overview, on page 11.

Apple Push Notifications

Notification Configuration

To configure ANDSF to send a message to a subscriber with an Apple iPhone or other iOS device, perform the following steps:

Step 1  Login to Policy Builder.
Step 2  Go to Reference Data > Systems > a system or a cluster > Plugin Configurations > Notification Configuration.
Step 3  Click the check box next to Apple Push Notification Configuration.
Step 4  View the Apple Push Notification Configuration pane that drops down.
The following parameters can be configured under Apple Push Notification Configuration:

**Table 23: Apple Push Notification Configuration Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| APNS Server Address        | • Apple Production: Connects to gateway.push.apple.com on port 2195.  
• Apple Test: Connects to gateway.sandbox.push.apple.com on port 2195.  
• Other: Uses a server address other than the standard Apple ones. It also uses the other gateway and server port fields defined in Other Server Gateway and Other Server Port fields below. |
| Other Server Gateway       | Name of a server gateway if Other is selected as APNS Server Address.                                                                                                                                     |
| Other Server Port          | Port number of the server gateway if Other is selected as APNS Server Address.                                                                                                                             |
| Certificate                | This certificate is used to make a secure connection with APNS server. You must provide a certificate file that is loaded into ANDSF. The certificate should be in .p12 format.                          |
| Certificate Password       | Password added to the certificate when connecting to the APNS.                                                                                                                                             |

**Step 5**

Go to Message Configuration, on page 91 to configure the message to be sent for the notification configuration done above.
**Message Configuration**

To create the messages for a subscriber’s Apple iPhone or the Apple iOS operating system to be sent by ANDSF, perform the following steps:

**Step 1**  Select Reference Data tab > Notifications > Apple Push Notifications.

**Step 2**  From right side, click Apple Push Notification under Create Child to open the pane as shown below:

*Figure 73: Apple Push Notification*

The following parameters can be configured under Apple Push Notification:

*Table 24: Apple Push Notification Message Configuration Parameters*

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the notification message. This name is used in the action phrase in the policy definition. Best practice is to make this short, but meaningful and unique.</td>
</tr>
<tr>
<td>Badge</td>
<td>Default is 0 (number). The number to display as the badge of the Apple Push Notification icon. If this property is absent, the badge is not changed. To remove the badge, set the value of this property to 0. Example: 1, 2, 3, ....</td>
</tr>
</tbody>
</table>
### GCM Notifications

The Google Cloud Messaging (GCM) notification service provides a simple, lightweight mechanism that servers can use to tell mobile applications to contact the server directly, to fetch updated application or user data.

#### Notification Configuration

**Step 1**  
Login to Policy Builder.

**Step 2**  
Go to **Reference Data > Systems > a system or a cluster > Plugin Configurations > Notification Configuration**.

**Step 3**  
Click the check box next to Google Cloud Messaging Notification Configuration.

**Step 4**  
View the **Notification Configuration** screen that drops down.
The following parameters can be configured under Google Cloud Messaging Notification Configuration:

**Table 25: Google Cloud Messaging Notification Configuration Parameters**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Api Key</td>
<td>String - An API key that is saved on the 3rd-party app server that gives the app server authorized access to Google services. The API key is included in the header of POST requests. This is obtained from Google. For example, AIzaSyBrnMySrXz-FLkknbzI-1EZ47WiYWR4Zgs</td>
</tr>
<tr>
<td>Sender Id</td>
<td>Sender ID has to be pre-created as a part of GCM Registration process. For example, 114402260484</td>
</tr>
<tr>
<td>Delay while Idle</td>
<td>Boolean - When this parameter is set to true (checked), it indicates that the message should not be sent until the device becomes active. The default value is false (unchecked).</td>
</tr>
<tr>
<td>Time To Live Days</td>
<td>This parameter specifies how long (in seconds) the message should be kept in GCM storage if the device is offline. The maximum time to live supported is 4 weeks. The default value is 4 weeks.</td>
</tr>
<tr>
<td>Proxy</td>
<td>These are very basic and will need to be provided by the customer.</td>
</tr>
</tbody>
</table>
If both an XMPP and HTTP server are present, it will first try the XMPP server first. If that fails it will default to the HTTP.

Location of the XMPP Servers are defined in the IP Address of the XMPP Server List.
Location of the HTTP Server is defined in qns.conf:

\* Dgcm.send.endpoint.http

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCM XMPP Servers and HTTP Server</td>
<td>If both a XMPP and HTTP servers are present, it will first try the XMPP server first. If that fails it will default to the HTTP. Location of the XMPP Servers are defined in the IP Address of the XMPP Server List. Location of the HTTP Server is defined in qns.conf:</td>
</tr>
<tr>
<td>Ip Address</td>
<td>String - hostname or IP address of GCM XMPP server.</td>
</tr>
<tr>
<td>Port</td>
<td>Port number of the GCM XMPP server.</td>
</tr>
<tr>
<td>SSL Connection</td>
<td>Select the required SSL connection from the drop-down list. It determines if the connection uses TLS. The available values are Disabled, Enabled and Required.</td>
</tr>
<tr>
<td>Allow Self Signed Certificate</td>
<td>Boolean - By default, it is checked (true).</td>
</tr>
<tr>
<td>XMPP Domain</td>
<td>The name of the service provided by an GCM XMPP server.</td>
</tr>
<tr>
<td>Stream Compression</td>
<td>If checked, compression is used for the connection. By default it is checked (true).</td>
</tr>
<tr>
<td>Check Expired Certificates</td>
<td>If checked, XMPP server will look for expired certificates. By default, it is checked (true).</td>
</tr>
<tr>
<td>Lb Type</td>
<td>Select the required lb type from drop-down list. GCM Load Balancing Type: Active, Standby</td>
</tr>
<tr>
<td>Retries</td>
<td>This parameter can be set as per user needs of limiting message retransmission limit in case it has not delivered. Default value is 1.</td>
</tr>
</tbody>
</table>

Configure the GCM notification message to be sent for the notification configuration done above.

**Message Configuration**

To create the real-time notification to be sent by ANDSF, perform the following steps:

**Step 1**  Select **Reference Data > Notifications > GCM Notifications**.

**Step 2**  In the right panel, click **GCM Notification** under **Create Child**. The GCM Notification window is displayed.
The following parameters can be configured under GCM Notification:

**Table 26: GCM Notification Message Configuration Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>String - Message Name.</td>
</tr>
<tr>
<td>Collapse Key</td>
<td>String - This parameters identifies a group of messages (e.g., with collapse_key: “Updates Available”) that can be collapsed, so that only the last message gets sent when delivery can be resumed. This is intended to avoid sending too many of the same messages when the device comes back online or becomes active.</td>
</tr>
<tr>
<td>Time To Live Days</td>
<td>Integer - Overrides setting in the GCM Notifications Plug-in Configuration for this message.</td>
</tr>
<tr>
<td>Send Once Per Session</td>
<td>If checked, only one GCM Notification is sent once per session. By default, it is not checked - ANDSF server updates the UE for all changes in realtime.</td>
</tr>
<tr>
<td>Delay While Idle</td>
<td>If checked, it overrides setting in GCM Notifications Plug-in Configuration for this message.</td>
</tr>
</tbody>
</table>
To use GCM Notifications, we need to configure Service Options. For more information on the configuration, refer to Service Option Configuration.

**Service Option Configuration**

This section provides an example Service Options configuration which can be used for Apple Push and GCM notification. The bodies of the messages are identical to make the service options parameters simpler to follow.

**Use Case Template Configuration - APNS**

Figure 76: Use Case Template Configuration - APNS

### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Message | JSON or Plain text can be used for the templates. JSON must be a complete JSON document, not a partial document.  
For example: `{"json":{"id":"$id","some":"thing","someother":"$thing"},
"additional":"$replacement"}` is valid but "id": "$id", "some": "thing", "someother": "$thing", 
"additional": "$replacement" is not valid because it is not a complete document.  

**Note** If you are using a Chemring client, then use ANDSF_PULL as message to enable GCM Push Notification so that the UE sends a PULL request to the ANDSF Server. |
In **Notification to Send**, we can give the name of notification for Apple Push Notification. In **Override Destination Retriever**, select **Session UE APNS Registration Key**. This will retrieve APNS token from the subscriber session.

**Use Case Template - GCM**

**Figure 78: Use Case Template - GCM**
Service Configuration - GCM

Figure 79: Service Configuration - GCM

To configure a service for GCM Notification, set the following parameters:

- In Notification to Send, enter the name of notification for GCM Notification.

- In Override Destination Retriever, select Session UE GCM Registration Key. This will retrieve GCM token from the subscriber session.

- Specify the MessageParameter as described:
  - Code: $imsi
  - Value Retriever: Session IMSI
Domain Overview

A domain controls how a user is authorized.

Domain provides multiple advanced options which help us to take some default actions based on the conditions. Advanced rules determine if unknown subscribers can come into the system and defines the unknown service. This is often used if subscribers self-provision and so are initially unknown or a default service can be assigned to a known subscribers.

When multiple domains are configured it is very difficult to select a single domain to authorize/authenticate a subscriber. This problem can be overcome by configuring the Locations on the individual domains. Location provides an option to select the individual domain based on the attributes received from the incoming request like Framed-IP, NAS-IP or based on AVP with the combination of Time Zone.

USuM

Once a user is authorized, domains can also auto-provision a user in USuM (including a default Service). If a user is not auto-provisioned, the user must have been provisioned by API into USuM before they are assigned a Service on the network.

Each user goes through a single domain authorization process upon log in. There can be multiple domains configured each having different kind of authorization. A user's domain is determined by Location. If a user does not match any of the Domains, they are considered to be part of the Domain marked as 'default'.

A domain can also auto provision a subscriber in SPR and associate a default service to it. This provides an option to register the subscriber based on Primary Credential and Password received from the incoming request, for example, Radius Username and Radius Password. This method is generally used in scenarios where the system is configured to “auto-learn” subscribers and assign a default service profile.

External SPR

You can configure an external SPR that ANDSF can use to validate subscribers.

To use an external SPR for authorization, configure the URL of the external SPR in the ANDSF plugin configuration as described in ANDSF Configuration. The Policy Server uses the same URL to connect to the
external SPR. When you configure the domain for ANDSF, you must also configure the anonymous service that is attached to the subscriber so that the subscriber can be assigned a policy.

**Configure Domain for USuM**

In Policy Builder, navigate to Services > Create Domain.

In the General tab, select USuM Authorization. USuM Authorization method authenticates the subscriber based on the field selected at User Id Field. The following table describes the parameters that are displayed when you select USuM Authorization:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Id Field</td>
<td>Defines what field to use as userid while authorizing. By default, it is set to use Session Username.</td>
</tr>
<tr>
<td>Remote Db Lookup Key Field</td>
<td>Defines what field to use as remote lookup key when choosing remote database in USuM configuration. By default, it is left as blank.</td>
</tr>
<tr>
<td>Domain Prefix</td>
<td>It is used to give a prefix to the domain we are configuring and it will get appended to usernames.</td>
</tr>
<tr>
<td>Append Location</td>
<td>By selecting this check box, we can also partition username based on location. For example, &lt;location&gt;/&lt;username&gt;</td>
</tr>
</tbody>
</table>

Rest of the tabs are not required for ANDSF configuration.

**Configure Domain for External SPR**

In Policy Builder, navigate to Services > Create Domain.

In the General tab, select Allow All Users.

Select the Advanced Rules tab. Select an Anonymous Service to apply a service to users that map to this Domain.

Once you have configured the domain, you must also configure the service to use the external SPR. For more information, see Configure Lookup for External SPRs, on page 115.

For more information about Domains, see CPS Mobile Configuration Guide.
Services

Overview

In ANDSF, a 'Service' is what is assigned to a subscriber (in USuM) to define how that subscriber is treated. Some basic examples of services would be a 'GOLD' user might get a high upload/download speed whereas a 'BRONZE' user would get a low one. Other examples would include having one type of user be redirected to a portal when their Quota is exhausted whereas another type would only have their speed downgraded.

As the Service maps as closely as possible to how a Service Provider wants to classify their customers, the Service in ANDSF is flexibly defined to allow configuration at different levels.

Below is an overview of the different objects referenced in the Services tab in Policy Builder. The detailed description of each object is provided in below sections.

Figure 80: Service

- A service is effectively just a 'code' to label the service and a collection of Service Options which contain the definition of what a service 'is'.
- What a Customer Service Representative assigns to a subscriber to describe the user's plan.
- Multiple services can be assigned to a single subscriber
• If multiple services are assigned to a subscriber, the service options are combined between all assigned services. Therefore, there is no logical difference between a subscriber with:
  • A single service with 10 service options
  • 10 services with 1 option each

Service Option

• Provides the concrete values which can be re-used for multiple services.

• What values are configurable in a Service Option are setup by the Use Case Template object. The Use Case Template can provide defaults to the Service Option or hide values in Service Configuration objects not necessary for certain use cases.

• If a Service Configuration's value is not defined in a Service Option, the value from the Use Case Template will be used.

Service Configuration

• The low-level configuration objects is used by the ANDSF code to drive functionality. These objects are used to drive functionality in the system. The whole point of the Service > Service Option > Use Case Template chain of functionality is to flexibly configure these Service Configuration objects which the code uses to drive system logic.

• These objects are defined by the code.

• Types of service configurations:
  • PriorityConfiguration: Only 1 allowed to be active at a time. If multiples priority configurations are added, highest priority is used.
    These are used in cases where only a single value makes sense. For example, when sending an 'Accept' message, we can only have one template and multiples don't make sense.
    Objects of this type always have a priority field. If multiple priority configurations are added, the highest priority object is used.
    Example: AccessAcceptConfiguration, RegisterMacAddress
  • GroupConfiguration (most common): Only 1 per 'Group Name' are allowed to be active. If multiple configurations are added highest priority per 'Group Name' is used.
    These are used in cases where a configuration only makes sense for a single 'group' (key). For example, if it makes sense to control the upload/download speed based on the network type (cell, wifi and so on), a service configuration to control network speed with a group set for cell/wifi would allow multiple service configurations to be added.
    These objects always have a group field as well as a priority field. For each unique group value, the highest priority is used.

Use Case Template

• Defines the Service Configuration objects to be set by a Service Option and can provide default values and/or hide values which don't need to be set by a use case.
• Optionally contains 'Initiators' (Conditions) which define when the template is active.
• Created by an advanced user (usually Engineering/AS).
• Makes Service Option and Service creation easier.

**Use Case Option**

• A child of Use Case Template used to add/modify Service Configurations objects when certain conditions occur.
• Provides a way to separate Service Configurations within a use case based on conditions.
• Contains the same functionality of a Use Case Template.
• Can add new service options or modify service options from parent Use Case Template.

**Service Screens**

**Default ANDSF Service**

*Figure 81: Default ANDSF Service*

This is the default ANDSF service which retrieves static policy based on the following two Service Options:

• Andsf_DEF_NAMED
• Andsf_Location_OR
Service Options

Andsf_DEF_NAMED

This service option will lookup Named_Ext_Location_lookup (default :null) Name from Policy Extension Location lookup.

**Figure 82: Service Option - Andsf_DEF_NAMED**

Andsf_Location_OR

This service option will lookup TGPP Location or WLAN Location Name from Policy Extension Location lookup during provisioning. This also maps DevInfo and UELocation parameters to its URI Types.
### Service Option (Read Only)

**Name**  
Andsf_Location_OR

**Use Case Template:** Andsf_ISMP_LOC

#### Service Configurations

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Pull Value from...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Group Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANDES Client</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MO Type</td>
<td>ISMP</td>
<td></td>
</tr>
<tr>
<td>Request Type</td>
<td>Any</td>
<td></td>
</tr>
<tr>
<td>Lookup Type</td>
<td>Policy_EXT_LOCATION</td>
<td></td>
</tr>
</tbody>
</table>

#### LookupDM Provision Parameters

<table>
<thead>
<tr>
<th>Display Name</th>
<th>Value</th>
<th>Pull Value from...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Join By</td>
<td>OR</td>
<td></td>
</tr>
<tr>
<td>Key</td>
<td>ID</td>
<td></td>
</tr>
<tr>
<td>Match Type</td>
<td>Equals</td>
<td></td>
</tr>
<tr>
<td>Key</td>
<td>WLAN_Location (Policy_EXT_LOCATION)</td>
<td></td>
</tr>
</tbody>
</table>

### Actions

**Copy:**

- Current Service Option

#### Lookup Keys

<table>
<thead>
<tr>
<th>Match Type</th>
<th>Value</th>
<th>Pull Value from...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Join By</td>
<td>OR</td>
<td></td>
</tr>
<tr>
<td>Key</td>
<td>ID</td>
<td></td>
</tr>
</tbody>
</table>

Max Locations

Send URI  false
The Match Type for LookupKeys includes the values: EndsWith, Equals, RegEx (regular expression), and StartsWith. If you are using RegEx, you must not use the Match Type in Control Center. You can use RegEx to define a search pattern with wildcards. For example, the following image shows how you can use the RegEx in Control Center: .*silver|bronze.* In this example, ANDSF looks for MO Type that may include either silver or bronze in the name.

Figure 84: RegEx Example in Control Center

Use Case Template

The following Use Case Templates are used to create the service options:

- Andsf_ISMP_DEF
- Andsf_ISMP_LOC
### Andsf_ISMP_DEF

**Figure 85: Use Case Template - Andsf_ISMP_DEF**

<table>
<thead>
<tr>
<th>Service Configurations</th>
<th>DefaultDMTProvision - Named Lookup Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>*Display Name</td>
</tr>
<tr>
<td>Priority</td>
<td>0</td>
</tr>
<tr>
<td>Group Name</td>
<td></td>
</tr>
<tr>
<td>ANDSF Client</td>
<td></td>
</tr>
<tr>
<td>MO Type</td>
<td>ISMP</td>
</tr>
<tr>
<td>Request Type</td>
<td>Any</td>
</tr>
<tr>
<td>DM Tree Lookup</td>
<td>Policy_EXT_LOCATION</td>
</tr>
<tr>
<td>DM Lookup Name</td>
<td>Named_EXT_Config_lookup</td>
</tr>
<tr>
<td>Send URI</td>
<td>False</td>
</tr>
</tbody>
</table>

### Andsf_ISMP_LOC

**Figure 86: Use Case Template - Andsf_ISMP_LOC**

<table>
<thead>
<tr>
<th>Service Configurations</th>
<th>LookupDMTProvision Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>*Display Name</td>
</tr>
<tr>
<td>Priority</td>
<td>0</td>
</tr>
<tr>
<td>Group Name</td>
<td></td>
</tr>
<tr>
<td>ANDSF Client</td>
<td></td>
</tr>
<tr>
<td>MO Type</td>
<td>ISMP</td>
</tr>
<tr>
<td>Request Type</td>
<td>Any</td>
</tr>
<tr>
<td>Lookup Type</td>
<td>Policy_EXT_LOCATION</td>
</tr>
<tr>
<td>Lookup By (List)</td>
<td></td>
</tr>
<tr>
<td>Max Locations</td>
<td></td>
</tr>
<tr>
<td>Send URI</td>
<td>False</td>
</tr>
</tbody>
</table>
This is the default override service where incoming TGPP and WLAN locations override existing values (based on Default Location Override Lookup).

Policy retrieval is based on the following service options:

- Andsf_ISMP_DEF_LOC_OVER

**Service Options**

**Andsf_ISMP_DEF_LOC_OVER**

This service option will lookup Default_Loc_Override UseCase Name from Policy Default Override Location lookup during provisioning.
The following Use Case Template is used to create the service options:

- Andsf_ISMP_DEF_LOC_OVERRIDE
Andsf-ISMP_DEF_LOC_OVERRIDE

Figure 89: Use Case Template

This is a location based service where incoming TGPP and WLAN Locations are used for policy lookup. Policy retrieval is based on the following service option:
• Andsf_Location_OR

Service Options

Refer to Andsf_Location_OR.

Use Case Template

Refer to Andsf_ISMP_LOC, Use Case Template, on page 106.

Tier based ANDSF Service

*Figure 91: Tier based ANDSF Service*

This is a tier based service where incoming TGPP and WLAN Locations, along with subscriber tier are used for policy lookup. Policy retrieval is based on the following service option:

• Andsf_Location_Tier

Service Options

This service option will lookup TGPP_Location, WLAN_Location and Tier UseCase Name from Policy Extension tier lookup during provisioning. It will also map DevInfo and UELocation parameters to its URI types.
Use Case Template

Refer to Andsf_ISMP_LOC, Use Case Template, on page 106.
Geo Location based ANDSF Service

Figure 93: Service - Andsf_ISMP_GEO_LOC_STATIC

This is a location based service where incoming Geo locations are used for policy lookup. Policy retrieval is based on following service option:

- Andsf_ISMP_GEO_LOC_STATIC

Service Options

This service option will lookup GEO_Location UseCase Name from Geo Location Based lookup during provisioning.
It will also map DevInfo and Geo_Location parameters to its URI types.

Use Case Template

The following Use Case Template is used to create this service option:

---

**Use Case Template**

<table>
<thead>
<tr>
<th>Display Name</th>
<th>Value</th>
<th>Pull Value from...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Group Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANDSF Client</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MO Type</td>
<td>ISMP</td>
<td></td>
</tr>
<tr>
<td>Request Type</td>
<td>Any</td>
<td></td>
</tr>
<tr>
<td>URI Type</td>
<td>UELocation_GEO_LOC</td>
<td>Policy_EXM_GEO_LOC_STATIC</td>
</tr>
<tr>
<td>Get From Device If Missing</td>
<td>true</td>
<td></td>
</tr>
</tbody>
</table>

---
**Configure Lookup for External SPRs**

Create a service as per your requirements and configure a Lookup for External SPR on the basis of the feature list coming from the External SPR Response:

In Policy Builder **Reference Data > DM configuration**, configure a key named Feature in the Lookup. Bind this value to the Session Policy Field `<ANDSF External SPR feature>`.

In Control Center, add the actual lookup. Add the appropriate fields and the DM group as required.

**Configure Lookup for Extended Device Information**

You can set up ANDSF server to facilitate the dynamic configuration of lookup keys based on any entry in the DevInfo MO. You can then use this configuration to create and deliver policies based on extended Device Information in general.

The following example illustrates how to configure lookup based on battery level of the device.

**Policy Builder**

Perform the following steps in Policy Builder:

1. Configure CRD for the new data type: Add a CRD table for the new URI to be added on the basis of which the lookup needs to be designed.
2. Configure URI Type: Add the new URI in the DM Configuration and provide the reference to the CRD that you have configured.

3. Configure the new lookup with the new URI binding.

4. Configuring the new Service for the New Client URI: Add the following service configuration in the Use Case so that the Lookup can configure the new URI.
Control Center

Perform the following steps in Policy Builder:

1. Configure the CRD with the value from the UE.
2. Configure the new lookup according to the new URI Values.

You can verify the URI processing in the qns.log files. The following log file entry is a sample:

INFO : (ANSDF) DevInfo processed : vendor NA DevId: US14525_Test_1 DevType: samsung
INFO : (ANSDF) Processed URI ./DevInfo value: [smartswitch]
INFO : (ANSDF) Processed URI ./DevInfo/Ext/UEClientVendor value: [ChemringTMO]
INFO : (ANSDF) checking state: RESP_GET {40}
INFO : (ANSDF) checking state: LOOKUP {90}
INFO : (ANSDF) Processing lookup Policy_EXT_ClientVendor
INFO : (ANSDF) Lookup using TIER value: [Gold]
INFO : (ANSDF) Lookup using ./DevInfo/Ext/UEClientVendor value: [ChemringTMO]
INFO : (ANSDF) Lookup Group:NewGroup