

CUDB SOAP Interwork Description

INTERWORK DESCR

Copyright

© Ericsson AB 2016-2018. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

Trademark List

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark Information.



Contents

1	Introduction	1
1.1	Document Purpose and Scope	1
1.2	Revision Information	1
1.3	Typographic Conventions	1
2	Overview	1
3	SOAP Protocol Introduction	2
3.1	SOAP Message Structure	2
3.2	Data Encoding	3
4	Notifications Interface	3
4.1	XML Data Structure	3
4.2	Bindings	8
4.3	SOAP Socket Initialization	9
5	WSDL Schema	10
6	Messages Examples	13
6.1	Successful Notification Messages	13
6.2	Failed Notification Messages	17
	Glossary	19
	Reference List	21





1 Introduction

This document is a description of the Simple Object Access Protocol (SOAP) based notification interface, that is the interface used for notifying changes in certain data stored in the Ericsson Centralized User Database (CUDB) to a generic application Front End (FE).

1.1 Document Purpose and Scope

The document describes the SOAP interface used by the CUDB in the notifications towards a generic application FE.

For more information about CUDB Notification function, refer to *CUDB Notifications*, Reference [1].

1.2 Revision Information

Rev. A

This document is based on 3/15519-HDA 104 03/9 and contains editorial changes only.

1.3 Typographic Conventions

Typographic conventions can be found in the following document:

- *Typographic Conventions*.

2 Overview

This document describes the notifications interface:

- Definition of the notifications web service and the Web Services Description Language (WSDL) schema.
- Description on how the SOAP client (CUDB) handles the notifications towards the SOAP servers (that is towards the application FEs): types of



SOAP socket initialization, message structures provided by the notification interface, as well as the parameters and their data.

SOAP makes use of Hypertext Transfer Protocol (HTTP) as an internet application layer protocol as a transport protocol. SOAP may also be used over Hypertext Transfer Protocol Secure (HTTPS), with either simple or mutual authentication. For the use of SOAP over HTTPS, CUDB must be carefully configured for using Transport Layer Security/Secure Sockets Layer (TLS/SSL). Refer to *CUDB Security and Privacy Management*, Reference [2] for further details on how to configure CUDB for TLS/SSL.

Note: Further occurrences of HTTP refer to HTTP or HTTPS when configured.

3 SOAP Protocol Introduction

The notification interface is based on SOAP 1.1 ([Simple Object Access Protocol \(SOAP\) 1.1, 2000-08, W3C SOAP/1.1 Note](#), Reference [5]). This chapter is a rough description of the SOAP standard. SOAP is a communication protocol between applications that provides a simple and lightweight mechanism for exchanging structured and typed information in a decentralized and distributed environment based on Extensible Markup Language (XML).

3.1 SOAP Message Structure

A SOAP message is an XML document that consists of a mandatory SOAP envelope, an optional SOAP header, and a mandatory SOAP body.

- **SOAP Envelope:** Root element of the message. It identifies the XML document as a SOAP message.
- **SOAP Header:** Mechanism used to add features to a SOAP message in a decentralized manner without prior agreement between the communicating parties.
- **SOAP Body:** A container for mandatory information intended for the ultimate recipient of the message. Inside the SOAP Body a SOAP Fault element can be included to carry error and/or status information.

When using SOAP in HTTP, the HTTP request/response message model is followed as defined in [Hypertext Transfer Protocol -- HTTP/1.1, 1999-06, RFC 2616](#), Reference [6], providing SOAP request parameters in an HTTP request and SOAP response parameters in an HTTP response.

The HTTP `content-type` entity header must be always set to `text/xml`.



The `SOAPAction` HTTP request header field is received as part of the HTTP request, and its value is `notify`.

SOAP HTTP follows the semantics of the HTTP Status codes defined in [Hypertext Transfer Protocol -- HTTP/1.1, 1999-06, RFC 2616](#), Reference [6] for communicating status information in HTTP.

3.2 Data Encoding

The data in the notifications is encoded in a format that is not tied to the application data model. This means there are no exclusive data types for a specific application FE. The Lightweight Directory Access Protocol (LDAP) ([Lightweight Directory Access Protocol \(LDAP\): Directory Information Models, RFC 4512](#), Reference [7]) attributes in the message are encoded as either of the following:

- Character strings for non-binary attributes.
- Base64-encoded strings for binary attributes.

CUDB is aware of the data model and translates from the original data type to string type.

4 Notifications Interface

This section describes the XML data structure, the HTTP structure, and the general procedures.

Note: XML elements and attributes are case sensitive.

4.1 XML Data Structure

4.1.1 Envelope XML Elements

The information to be included in the `SOAP Envelope` is specified in [Simple Object Access Protocol \(SOAP\) 1.1, 2000-08, W3C SOAP/1.1 Note](#), Reference [5]. It is mandatory to fill the `SOAP Envelope`.

4.1.2 Header XML Elements

There is no specific information to be included in the `SOAP header`. The `SOAP header` is optional and it is not sent in the notification.

4.1.3 Body XML Elements

4.1.3.1 Notifications Requests

The root of the SOAP Body for Notifications Request messages is a `notifyRequest` XML element which can be composed of the following XML elements:

- `notificationEvent`
- `additionalInformation`
- `modificationInformation`

Figure 1 shows a graphic diagram of the SOAP Body XML elements and attributes.

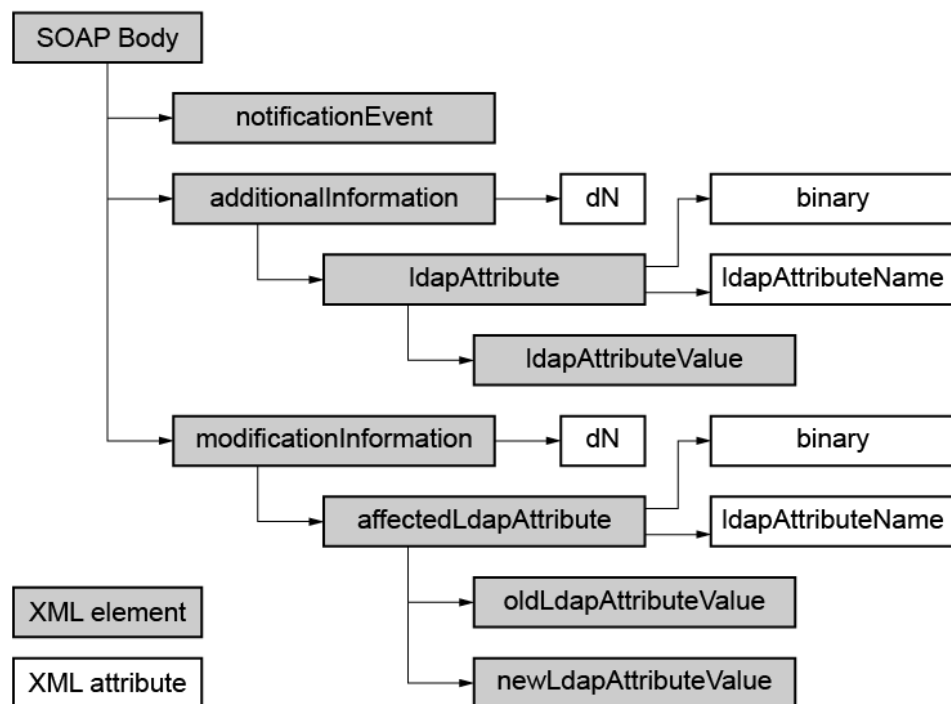


Figure 1 SOAP Body XML Elements and Attributes

4.1.3.1.1 notificationEvent

The `notificationEvent` is an XML string (refer to [Simple Object Access Protocol \(SOAP\) 1.1, 2000-08, W3C SOAP/1.1 Note](#), Reference [5]) that indicates the purpose of the notification. This XML element is optional (although it is always sent in the notification) and it is string data type.



It takes the value defined in `notificationString` attribute of the `CudbNotificationEvent` class in the configuration, refer to *CUDB Node Configuration Data Model Description*, Reference [3] for further details on CUDB configuration.

4.1.3.1.2 `additionalInformation`

The `additionalInformation` XML element is a structure providing relevant information for the handling of the notification.

There may be multiple instances of this XML element present in the notification request.

The `additionalInformation` XML element is composed of a series of `ldapAttribute` XML elements, an Extension XML element (included to provide future extensibility), and the XML attribute as shown in Table 1.

Table 1 `additionalInformation` XML Attributes

XML attributes	Type	Required	Description
<code>dn</code>	String	Mandatory	Stores the provisioned Distinguished Name (DN) of the LDAP data object.

The **`ldapAttribute`** XML element is a structure representing an LDAP attribute related to the user subject of the notification event. Table 2 represents the XML attributes that can be set to this XML element.

Table 2 `ldapAttribute` XML Attributes

XML attributes	Type	Required	Description
<code>binary</code>	Boolean	Optional	For <code>binary</code> LDAP attributes (whose value is encoded in Base64), its value is "true" or "yes". The default value (to be assumed if the XML attribute is not present) is "false" or "no".
<code>ldapAttribute</code> <code>name</code>	String	Mandatory	Stores the name of the LDAP attribute.

The **`ldapAttribute`** datatype is composed of a series of **`ldapAttributeValue`** XML elements (data type string) containing the value of the attribute. For multivalued LDAP attributes there is a series of `ldapAttributeValue`. The series of `ldapAttributeValue` elements include all the values of the LDAP attribute.



4.1.3.1.3 modificationInformation

The **modificationInformation** XML element is a structure representing the LDAP object instance whose modification triggered the notification event. This XML element is mandatory for `ProvisioningEvent` notifications. When present, there may be multiple instances of this XML element within a Notifications Request message.

It is composed of a series of `affectedLDAPAttributes` XML elements, an Extension XML element (included to provide future extensibility), and the XML attributes as shown in Table 3.

Table 3 *modificationInformation XML Attributes*

XML attributes	Type	Required	Description
dN	String	Mandatory	Stores the Distinguish Name of the LDAP data object. If the dN can not be associated to an LDAP object instance of the application FE Directory Information Tree, the processing of the modification of this object shall be skipped.

affectedLDAPAttribute

The `affectedLDAPAttribute` XML element is a structure representing the LDAP attribute whose modification triggered the event. Table 4 represents the XML attributes that can be set to this XML element.

Table 4 *affectedLDAPAttribute XML Attributes*

XML attributes	Type	Required	Description
binary	Boolean	Optional	For binary LDAP attributes (whose value is encoded in Base64), its value is "true" or "yes". The default value (to be assumed if the XML attribute is not present) is "false" or "no".
ldapAttributeName	String	Mandatory	Stores the name of the LDAP attribute. If the attribute is not related to the corresponding LDAP object class associated to the application FE, the processing of the modification of this attribute shall be skipped.

The `affectedLDAPAttribute` is composed of an ordered sequence of:

- a series of `oldAttributeValue` XML elements.
- a series of `newAttributeValue` XML elements.

oldAttributeValue

XML string storing the attribute value before the modification took place.

newAttributeValue

XML string storing the attribute value after the modification took place. It is not sent if previously it did not exist any value before the operation.

Note: For multivalued LDAP attributes there can be a series of `oldAttributeValue` and `newAttributeValue` elements. The series of `oldAttributeValue` elements will include all the values of the LDAP attribute previous to the operation. The series of `newAttributeValue` elements will include all the values of the LDAP attribute after the operation.



4.1.3.2 Notifications Responses

For successful Notifications Response messages, the root of the SOAP Body is an empty `notifyResponse` XML element.

For unsuccessful Notifications Response messages, the root of the SOAP Body is a SOAP Fault element as defined in [Simple Object Access Protocol \(SOAP\) 1.1, 2000-08, W3C SOAP/1.1 Note](#), Reference [5].

4.2 Bindings

The only supported binding to carry the SOAP messages is the HTTP POST message request and answer ([Hypertext Transfer Protocol -- HTTP/1.1, 1999-06, RFC 2616](#), Reference [6]).

4.2.1 SOAP HTTP Request

The general format of the HTTP request defined in ABNF ([Augmented BNF for Syntax Specifications: ABNF, 1997-11, RFC 2234](#), Reference [8]) is the following:

```
Request=      Request-Line= POST SP requested-web-service
               SP HTTP/1.1
               *(( general-header
                  | request-header
                  | entity-header ) CRLF)
               CRLF
               [ message-body ]
```

requested-web-service contains the Universal Resource Locator (URL)-path of the SOAP server managing the notifications request.

For more information about the possible HTTP headers to be defined in the request, refer to [Hypertext Transfer Protocol -- HTTP/1.1, 1999-06, RFC 2616](#), Reference [6].

The information to be included in the different HTTP headers to carry a SOAP message in the HTTP request is specified in [Simple Object Access Protocol \(SOAP\) 1.1, 2000-08, W3C SOAP/1.1 Note](#), Reference [5]. The `SOAPAction` HTTP request header field (specific for SOAP) is defined as mandatory in [Simple Object Access Protocol \(SOAP\) 1.1, 2000-08, W3C SOAP/1.1 Note](#), Reference [5], and its value is `notify`.



The message body contains a SOAP message including the envelope, the header and the body. The information to be included in each SOAP part is specified in Section 4.1 on page 3.

4.2.2 SOAP HTTP Answer

The general format of the HTTP answer defined in ABNF ([Augmented BNF for Syntax Specifications: ABNF, 1997-11, RFC 2234](#), Reference [8]) is the following:

```
<Response>:: <Status-Line: HTTP/1.1 SP Status-Code SP Reason-Phrase>
=
  *(( general-header
    | request-header
    | entity-header ) CRLF)
  CRLF
  [ message-body ]
```

For more information about the possible HTTP headers to be defined in the request, refer to [Hypertext Transfer Protocol -- HTTP/1.1, 1999-06, RFC 2616](#), Reference [6].

The message body contains a SOAP message including the envelope, the header and the body. The information to be included in each SOAP part is specified in Section 4.1 on page 3.

The following HTTP Status codes defined in [Hypertext Transfer Protocol -- HTTP/1.1, 1999-06, RFC 2616](#), Reference [6], with their corresponding Reason-Phrase are sent by the application FE:

200 OK and empty **notifyResponse** XML element, when the operation was successful.

500 Internal Server Error with the corresponding SOAP Fault Code (either *Client* or *Server*) as defined in [Simple Object Access Protocol \(SOAP\) 1.1, 2000-08, W3C SOAP/1.1 Note](#), Reference [5].

Other Status Codes related to HTTP protocol errors could be also sent by the application FE. For more information, refer to [Hypertext Transfer Protocol -- HTTP/1.1, 1999-06, RFC 2616](#), Reference [6]. In these cases, no SOAP message is included.

4.3 SOAP Socket Initialization

The SOAP API provides two different mechanisms to manage the sockets opened between clients and endpoints:



- **KeepAliveIO:** TCP connections are reused and not closed when notifications are sent towards the SOAP server. Works on HTTP and TCP levels.
- **Non KeepAliveIO:** TCP connections are opened and closed once notifications are sent.

CUDB uses by default the `KeepAliveIO` mechanism for all the configured notification events. Socket management mechanism can be configured per event by means of the `soapInitKeepalive` attribute of the `CudbNotificationEvent` class in the configuration.

Note: Configuring `KeepAliveIO` is restricted to Ericsson personnel. Contact the next level of maintenance support to perform such procedures.

5 WSDL Schema

The notifications are built according to the following WSDL schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:notificationtypes=
    "http://schemas.ericsson.com/udc/1.0/types/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://schemas.ericsson.com/udc/1.0/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="udc"
  targetNamespace="http://schemas.ericsson.com/udc/1.0/">
  <wsdl:types>
    <xsd:schema attributeFormDefault="qualified"
      elementFormDefault="unqualified"
      targetNamespace="http://schemas.ericsson.com/udc/1.0/types/"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <xsd:complexType name="LdapAttributeType">
        <xsd:sequence>
          <xsd:element maxOccurs="unbounded"
            name="ldapAttributeValue" type="xsd:string"/>
        </xsd:sequence>
        <xsd:attribute name="ldapAttributeName"
          type="xsd:string" use="required"/>
        <xsd:attribute name="binary" type="xsd:boolean"/>
      </xsd:complexType>
      <xsd:complexType name="ExtensionType">
        <xsd:sequence>
          <xsd:any maxOccurs="unbounded" minOccurs="0"
            namespace="##any" processContents="lax"/>
        </xsd:sequence>
      </xsd:complexType>
    </wsdl:types>
  </wsdl:definitions>
```



```

    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="AdditionalInformationType">
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" name="ldapAttribute"
        type="notificationtypes:LdapAttributeType"/>
      <xsd:element minOccurs="0" name="extension"
        type="notificationtypes:ExtensionType"/>
    </xsd:sequence>
    <xsd:attribute name="dN" type="xsd:string" use="required"/>
  </xsd:complexType>
  <xsd:simpleType name="OperationType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="create"/>
      <xsd:enumeration value="modify"/>
      <xsd:enumeration value="delete"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="ModificationType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="replace"/>
      <xsd:enumeration value="add"/>
      <xsd:enumeration value="remove"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:complexType name="AffectedLdapAttributeType">
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0"
        name="oldLdapAttributeValue" type="xsd:string"/>
      <xsd:element maxOccurs="unbounded" minOccurs="0"
        name="newLdapAttributeValue" type="xsd:string"/>
    </xsd:sequence>
    <xsd:attribute name="ldapAttributeName"
      type="xsd:string" use="required"/>
    <xsd:attribute name="modification"
      type="notificationtypes:ModificationType"/>
    <xsd:attribute name="binary" type="xsd:boolean"/>
  </xsd:complexType>
  <xsd:complexType name="ModificationInformationType">
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded"
        name="affectedLdapAttribute"
        type="notificationtypes:AffectedLdapAttributeType"/>
      <xsd:element minOccurs="0" name="extension"
        type="notificationtypes:ExtensionType"/>
    </xsd:sequence>
    <xsd:attribute name="dN" type="xsd:string" use="required"/>
    <xsd:attribute name="operation"
      type="notificationtypes:OperationType"/>
  </xsd:complexType>
  <xsd:complexType name="NotificationRequestType">

```



```
<xsd:sequence>
  <xsd:element minOccurs="0" name="notificationEvent"
    type="xsd:string"/>
  <xsd:element maxOccurs="unbounded" minOccurs="0"
    name="additionalInformation"
    type="notificationtypes:AdditionalInformationType"/>
  <xsd:element maxOccurs="unbounded" minOccurs="0"
    name="modificationInformation"
    type="notificationtypes:ModificationInformationType"/>
  <xsd:element minOccurs="0" name="extension"
    type="notificationtypes:ExtensionType"/>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="NotificationResponseType"/>
<xsd:element name="notify"
  type="notificationtypes:NotificationRequestType"/>
<xsd:element name="notifyResponse"
  type="notificationtypes:NotificationResponseType"/>
</xsd:schema>
</wsdl:types>
<wsdl:message name="notifyRequest">
  <wsdl:part name="request"
    element="notificationtypes:notify"/>
</wsdl:message>
<wsdl:message name="notifyResponse">
  <wsdl:part name="response"
    element="notificationtypes:notifyResponse"/>
</wsdl:message>
<wsdl:portType name="NotificationService">
  <wsdl:operation name="notify">
    <wsdl:input message="tns:notifyRequest"/>
    <wsdl:output message="tns:notifyResponse"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="NotificationServiceBinding"
  type="tns:NotificationService">
  <soap:binding style="document"
    transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="notify">
    <soap:operation soapAction="notify"/>
    <wsdl:input>
      <soap:body parts="request" use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body parts="response" use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:service name="NotificationService">
  <!-- WSDL ports and location to be
    configured in client and servers-->
```




```
</wsdl:service>
</wsdl:definitions>
```

Note: According to the WSDL schema, the types “Operation Type” and “Modification Type” are not mandatory to be sent in the SOAP messages. CUDB SOAP messages do not include such information about the LDAP operation that triggered the notification.

6 Messages Examples

This section provides examples of successful and failed notification messages.

6.1 Successful Notification Messages

6.1.1 Successful Notification towards HLR-FE

The following example shows a successful mobility notification request and response, sent towards a Home Location Register FE (HLR-FE).

The request message is the following:

```
POST / HTTP/1.1
Host: 127.0.0.1:8080
User-Agent: gSOAP/2.7
Content-Type: text/xml; charset=utf-8
Content-Length: 1229
Connection: close
SOAPAction: "notify"

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ns1="http://schemas.ericsson.com/udc/1.0/"
  xmlns:ns2="http://schemas.ericsson.com/udc/1.0/types/">
  <SOAP-ENV:Body>
    <ns2:notify>
      <notificationEvent>mobilityEvent</notificationEvent>
      <additionalInformation ns2:dN="EpsDynInfId=EpsDynInf,
        EpsStaInfId=EpsStaInf,serv=eps, mscId=9099999999000000,
```



```

ou=multiscs,ou=ft,o=cudb,c=es">
  <ldapAttribute
    ns2:ldapAttributeName="EpsMobilityNotifInfo">
      <ldapAttributeValue>4</ldapAttributeValue>
    </ldapAttribute>
  </additionalInformation>
</additionalInformation ns2:dN="serv=csp,
mscId=9099999999000000, ou=multiscs,ou=ft,o=cudb,c=es">
  <ldapAttribute ns2:ldapAttributeName="CDC">
    <ldapAttributeValue>3</ldapAttributeValue>
  </ldapAttribute>
  <ldapAttribute ns2:ldapAttributeName="IMSI">
    <ldapAttributeValue>10999999000000</ldapAttributeValue>
  </ldapAttribute>
  <ldapAttribute ns2:ldapAttributeName="PSLOC">
    <ldapAttributeValue>5</ldapAttributeValue>
  </ldapAttribute>
</additionalInformation>
</ns2:notify>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

The successful response message is the following:

```

HTTP/1.1 200 OK
Date: Sat, 09 Jun 2001 15:01:55 GMT
Server: Apache/1.3.14 (Unix) tomcat/1.0 PHP/4.0.1pl2
SOAPServer: SOAP::Lite/Perl/0.50
Cache-Control: s-maxage=60, proxy-revalidate
Content-Type: text/xml
Content-Length:427

```

```

<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ns2="http://schemas.ericsson.com/udc/1.0/types/">
  <SOAP-ENV:Body>
    <ns2:notifyResponse/>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

6.1.2

Successful Notification towards HSS/SLF-FE

The following example shows a successful mobility notification request and response, sent towards a Home Subscriber Server (HSS)/Subscription Locator Function (SLF) FE.



The request message is the following:

```
POST / HTTP/1.1
Host: 127.0.0.1:8080
User-Agent: gSOAP/2.7
Content-Type: text/xml; charset=utf-8
Content-Length: 1355
Connection: close
SOAPAction: "notify"

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ns1="http://schemas.ericsson.com/udc/1.0/"
  xmlns:ns2="http://schemas.ericsson.com/udc/1.0/types/">
  <SOAP-ENV:Body>
    <ns2:notify>
      <notificationEvent>mobilityEvent</notificationEvent>
      <additionalInformation ns2:dN="EpsDynInfId=EpsDynInf,
        EpsStaInfId=EpsStaInf,serv=eps, mscId=9099999999000000,
        ou=multiscs,ou=ft,o=cudb,c=es">
        <ldapAttribute ns2:ldapAttributeName="EpsMmeOdb">
          <ldapAttributeValue>1</ldapAttributeValue>
        </ldapAttribute>
        <ldapAttribute ns2:ldapAttributeName="EpsLocState">
          <ldapAttributeValue>1</ldapAttributeValue>
        </ldapAttribute>
        <ldapAttribute ns2:ldapAttributeName="EpsMmeAddr">
          <ldapAttributeValue>mme.ericsson.1.se
        </ldapAttributeValue>
        </ldapAttribute>
        <ldapAttribute ns2:ldapAttributeName="EpsMmeRealm">
          <ldapAttributeValue>epc.mnc015.mcc234.3gppnetwork.org
        </ldapAttributeValue>
        </ldapAttribute>
      </additionalInformation>
      <additionalInformation ns2:dN="serv=csp,
        mscId=9099999999000000,ou=multiscs,ou=ft,o=cudb,c=es">
        <ldapAttribute ns2:ldapAttributeName="SGSNNUM">
          <ldapAttributeValue>3</ldapAttributeValue>
        </ldapAttribute>
      </additionalInformation>
    </ns2:notify>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The response message is equal to the response in HLR-FE notifications, which is shown in Section 6.1.1 on page 13.

6.1.3 Successful Notification for a Modification of an Attribute

The following example shows a successful notification of when an attribute is modified:

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ns1="http://schemas.ericsson.com/udc/1.0/"
  xmlns:ns2="http://schemas.ericsson.com/udc/1.0/types/">
  <SOAP-ENV:Body>
    <ns2:notify>
      <notificationEvent>sampleEvent</notificationEvent>
      <additionalInformation ns2:dN="serv=csp,
        mscId=9099999999000000,ou=multiSCs,ou=ft,o=cudb,c=es">
        <ldapAttribute ns2:ldapAttributeName="UV">
          <ldapAttributeValue>83</ldapAttributeValue>
        </ldapAttribute>
      </additionalInformation>
      <modificationInformation ns2:dN="serv=csp,
        mscId=9099999999000000,ou=multiSCs,ou=ft,o=cudb,c=es">
        <affectedLdapAttribute ns2:ldapAttributeName="MV">
          <oldLdapAttributeValue>25</oldLdapAttributeValue>
          <oldLdapAttributeValue>30</oldLdapAttributeValue>
          <oldLdapAttributeValue>40</oldLdapAttributeValue>
          <newLdapAttributeValue>26</newLdapAttributeValue>
          <newLdapAttributeValue>45</newLdapAttributeValue>
        </affectedLdapAttribute>
      </modificationInformation>
    </ns2:notify>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

6.1.4 Successful Notification for a Modification of a Binary Attribute

The following example shows a successful notification of when a binary attribute is configured as a related notification.

```
<?xml version="1.0" encoding="UTF-8"?><SOAP-ENV:Envelope⇒
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" ⇒
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ⇒
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" ⇒
```



```

    xmlns:ns1="http://schemas.ericsson.com/udc/1.0/"
    xmlns:ns2="http://schemas.ericsson.com/udc/1.0/types/">
<SOAP-ENV:Body>
<ns2:notify>
<notificationEvent>NotifEvent</notificationEvent>
<additionalInformation ns2:dN="serv=fctest,mscId=9099999999001000,⇒
ou=multiSCs,ou=ft,o=cudb,c=es">
<ldapAttribute ns2:binary="true"⇒
  ns2:ldapAttributeName="FTESTATTRBINARY1REQUIREDMEMORY">
<ldapAttributeValue>QVFJRA==</ldapAttributeValue></ldapAttribute>
</additionalInformation>
</ns2:notify>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

6.2 Failed Notification Messages

This example shows an **HTTP 500 Internal Server Error** server response with SOAP Fault Code set to *Server*.

```

HTTP/1.1 500 Internal Server Error
Date: Sat, 09 Jun 2001 15:01:55 GMT
Server: Apache/1.3.14 (Unix) tomcat/1.0 PHP/4.0.1pl2
SOAPServer: SOAP::Lite/Perl/0.50
Cache-Control: s-maxage=60, proxy-revalidate
Content-Type: text/xml
Content-Length:611

```

```

<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ns2="http://schemas.ericsson.com/udc/1.0/types/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode xsi:type="xsd:string">SOAP-ENV:Server</faultcode>
      <faultstring xsi:type="xsd:string">
        Lack of resources in the server.
      </faultstring>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```





Glossary

For the terms, definitions, acronyms and abbreviations used in this document, refer to *CUDB Glossary of Terms and Acronyms*, Reference [4].





Reference List

CUDB Documents

- [1] *CUDB Notifications*
- [2] *CUDB Security and Privacy Management*
- [3] *CUDB Node Configuration Data Model Description*
- [4] *CUDB Glossary of Terms and Acronyms*

Other Documents and Online References

- [5] *Simple Object Access Protocol (SOAP) 1.1, 2000-08, W3C SOAP/1.1 Note* <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- [6] *Hypertext Transfer Protocol -- HTTP/1.1, 1999-06, RFC 2616* <http://www.rfc-editor.org/rfc/rfc2616.txt>
- [7] *Lightweight Directory Access Protocol (LDAP): Directory Information Models, RFC 4512* <http://www.rfc-editor.org/rfc/rfc4512.txt>
- [8] *Augmented BNF for Syntax Specifications: ABNF, 1997-11, RFC 2234* <http://www.rfc-editor.org/rfc/rfc2234.txt>