



A MITEL
PRODUCT
GUIDE

OpenScape Voice V11

Interface Manual: UM SPML

Interface Guide

04/2025

Notices

The information contained in this document is believed to be accurate in all respects but is not warranted by Mitel Europe Limited. The information is subject to change without notice and should not be construed in any way as a commitment by Mitel or any of its affiliates or subsidiaries. Mitel and its affiliates and subsidiaries assume no responsibility for any errors or omissions in this document. Revisions of this document or new editions of it may be issued to incorporate such changes. No part of this document can be reproduced or transmitted in any form or by any means - electronic or mechanical - for any purpose without written permission from Mitel Networks Corporation.

Trademarks

The trademarks, service marks, logos, and graphics (collectively "Trademarks") appearing on Mitel's Internet sites or in its publications are registered and unregistered trademarks of Mitel Networks Corporation (MNC) or its subsidiaries (collectively "Mitel"), Unify Software and Solutions GmbH & Co. KG or its affiliates (collectively "Unify") or others. Use of the Trademarks is prohibited without the express consent from Mitel and/or Unify. Please contact our legal department at jplegal@mitel.com for additional information. For a list of the worldwide Mitel and Unify registered trademarks, please refer to the website: <http://www.mitel.com/trademarks>.

© Copyright 2024, Mitel Networks Corporation

All rights reserved

Contents

History of Changes	3
1 UM SPML Interface Description	4
2 SPML Capabilities	4
3 Design Aspects	21
4 Request Templates	22

History of Changes

Issue	Date	Summary
1	07/2021	First issue of the guide.
2	11/2021	OCMP-5810
3	05/2024	Rebranding

1 UM SPML Interface Description

UM User Templates allow for creation of resources (Phones, UC Accounts, etc) based on user specific data. Bulk provisioning of Users and Resources by specifying user templates has been incorporated into LDAP Provisioning of UM Users. Starting with V8, User Template provisioning is introduced into CSV import and SPML Provisioning.

Since SPML provisioning is the interface provided from User Management to external applications (e.g. OSILA / OSAM) filtering/sorting/paging support is also introduced for Search Request, which is vital for large sets of Users.

1 SPML Capabilities

License & Permissions

Access to the SPML Interface of User Management shall be limited to admins that have the **OS CMP User Management Feature Package OSILA** profile assigned (and consumes one OS_UM_FP2 license).

SPML Request Types

The actions that can be performed through SPML Requests are the following:

2.1 Add

This kind of Request type is available only for the User Schema.

Example of SPML Add User Request:

```
<?xml version='1.0' encoding='UTF-8'?>
<s:Envelope xmlns:s='http://schemas.xmlsoap.org/soap/envelope/'>
<s:Body>
<spml:addRequest xmlns="urn:siemens:dxm:configuration:1:0"
xmlns:dsm1="urn:oasis:names:tc:DSML:2:0:core"
xmlns:spml="urn:oasis:names:tc:SPML:1:0"
```

```

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" requestID="add-
request">
<spml:operationalAttributes/>
<spml:identifier type="urn:oasis:names:tc:SPML:1:0#DN">
<spml:id>userid@system</spml:id>
<spml:identifierAttributes>
<spml:attr name="objectclass"><dsml:value
type="string">User</dsml:value></spml:attr>
<spml:attr name="domain"><dsml:value
type="string">system</dsml:value></spml:attr>
<spml:attr name="bulkactionid"><dsml:value
type="string">provisioning</dsml:value></spml:attr>
</spml:identifierAttributes></spml:identifier>
<spml:attributes>
<spml:attr name="loginName"><dsml:value
type="string">userid</dsml:value></spml:attr>
<spml:attr name="lastName"><dsml:value
type="string">lastName</dsml:value></spml:attr>
<spml:attr name="displayName"><dsml:value
type="string">displayName</dsml:value></spml:attr>
<spml:attr name="homeTimeZone"><dsml:value
type="string">America/Los_Angeles</dsml:value></spml:attr>
<spml:attr name="firstName"><dsml:value
type="string">firstNameCua</dsml:value></spml:attr>
<spml:attr name="language"><dsml:value
type="string">en_US</dsml:value></spml:attr>
<spml:attr name="businessPhone1"><dsml:value
type="string">+30210810898043</dsml:value></spml:attr>
<spml:attr name="fax"><dsml:value
type="string">21081870013</dsml:value></spml:attr>

```

```

<spml:attr name="notice"><dsml:value
type="string">notice</dsml:value></spml:attr>

<spml:attr name="emailAddress"><dsml:value
type="string">spmlUser@unify.com</dsml:value></spml:attr>

</spml:attributes>

</spml:addRequest>

</s:Body>

</s:Envelope>

```

NOTE: Not all the above fields are mandatory for creating a UM User.

2.2 Modify (where a defined attribute value can be modified but also deleted)

This kind of Request type is available only for the User Schema. An attribute can be manipulated through the “operation” selection, for instance, operation=“replace” will replace the old value of an attribute with a new one, while operation=“delete” will completely delete the value of an attribute.

Example of SPML Modify Request regarding the replacement of the values of the attributes *firstName* and *alternateEmailAddress* with new ones:

```

<?xml version='1.0' encoding='UTF-8'?>

<s:Envelope xmlns:s='http://schemas.xmlsoap.org/soap/envelope/'>

  <s:Body>

    <spml:modifyRequest xmlns="urn:siemens:dxm:configuration:1:0"
xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"
xmlns:spml="urn:oasis:names:tc:SPML:1:0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" requestID="modify-
request">

      <spml:identifier type="urn:oasis:names:tc:SPML:1:0#DN">

        <spml:id>userid@system</spml:id>

        <spml:identifierAttributes>

          <spml:attr name="objectclass"><dsml:value
type="string">User</dsml:value></spml:attr>

```

```

        <spml:attr name="domain"><dsml:value
type="string">system</dsml:value></spml:attr>

        <spml:attr name="bulkactionid"><dsml:value
type="string">provisioning</dsml:value></spml:attr>

    </spml:identifierAttributes>

</spml:identifier>

    <spml:modifications>

        <spml:modification name="firstName"
operation="replace"><dsml:value
type="string">firstNameNew</dsml:value></spml:modification>

        <spml:modification name="alternateEmailAddress"
operation="replace"><dsml:value
type="string">newemail@example.com</dsml:value></spml:modification>

    </spml:modifications>

</spml:modifyRequest>

</s:Body>

</s:Envelope>

```

Example of SPML Modify Request regarding the deletion of the value of the attribute emailAddress:

```

<?xml version='1.0' encoding='UTF-8'?>

<s:Envelope xmlns:s='http://schemas.xmlsoap.org/soap/envelope/'>

<s:Body>

<spml:modifyRequest xmlns="urn:siemens:dxm:configuration:1:0"
xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"
xmlns:spml="urn:oasis:names:tc:SPML:1:0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" requestID="modify-
request">

<spml:operationalAttributes>

<spml:attr name="updatemode"><dsml:value

```

```

type="string">UPDATE</dsml:value></spml:attr>
</spml:operationalAttributes>
<spml:identifier type="urn:oasis:names:tc:SPML:1:0#DN">
<spml:id>userid@system</spml:id>
<spml:identifierAttributes>
<spml:attr name="objectclass"><dsml:value
type="string">User</dsml:value></spml:attr>
<spml:attr name="domain"><dsml:value
type="string">system</dsml:value></spml:attr>
<spml:attr name="bulkactionid"><dsml:value
type="string">provisioning</dsml:value></spml:attr>
</spml:identifierAttributes>
</spml:identifier>
<spml:modifications>
<spml:modification name="emailAddress" operation="delete"><dsml:value
type="string"></dsml:value></spml:modification>
</spml:modifications>
</spml:modifyRequest>
</s:Body>
</s:Envelope>

```

Moreover, a User Template can also be applied to a UM User through SPML Request, by adding the operational attribute "userTemplateAssociationMode". The values that can be used are the following:

```

DO_NOT_APPLY_USER_TEMPLATE_RULES,
UNASSIGN_ADDITIONAL_RESOURCES,
UNASSIGN_DELETE_ADDITIONAL_RESOURCES,
UNASSIGN_PHONES_DELETE_OTHER_RESOURCES

```

Example of User Template application through SPML Modify Request:

```

<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <spml:modifyRequest xmlns="urn:siemens:dxm:configuration:1:0"
      xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"
      xmlns:spml="urn:oasis:names:tc:SPML:1:0"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" requestID="modify-
      request">
      <spml:operationalAttributes>
        <spml:attr name="updatemode">
          <dsml:value type="string">UPDATE</dsml:value>
        </spml:attr>
        <spml:attr name="userTemplateAssociationMode">
          <dsml:value
type="string">UNASSIGN_DELETE_ADDITIONAL_RESOURCES</dsml:value>
        </spml:attr>
      </spml:operationalAttributes>
      <spml:identifier type="urn:oasis:names:tc:SPML:1:0#DN">
        <spml:id>userid@system</spml:id>
        <spml:identifierAttributes>
          <spml:attr name="objectclass">
            <dsml:value type="string">User</dsml:value>
          </spml:attr>
          <spml:attr name="domain">
            <dsml:value type="string">system</dsml:value>
          </spml:attr>
          <spml:attr name="bulkactionid">
            <dsml:value type="string">provisioning</dsml:value>

```

```

        </spml:attr>
    </spml:identifierAttributes>
</spml:identifier>
<spml:modifications>
    <spml:modification name="userTemplate" operation="replace">
        <dsml:value type="string">utOSV_UC</dsml:value>
    </spml:modification>
    <spml:modification name="locality" operation="replace">
        <dsml:value type="string">OSVucLocality</dsml:value>
    </spml:modification>
    <spml:modification name="iMAddress" operation="replace">
        <dsml:value type="string">IM_Address</dsml:value>
    </spml:modification>
</spml:modifications>
</spml:modifyRequest>
</s:Body>
</s:Envelope>

```

2.3 Delete

This kind of Request type is available only for the User Schema.

NOTE: The user's resources can also be deleted using this request by setting the deleteResources attribute to true as shown in the following example. In case this attribute is omitted or set to false the resources will be unassigned.

Example of SPML Delete User Request:

```

<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
    <s:Body>
        <spml:deleteRequest xmlns="urn:siemens:dxm:configuration:1:0"

```

```

xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"
xmlns:spml="urn:oasis:names:tc:SPML:1:0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" requestID="delete-
request">

    <spml:operationalAttributes/>

    <spml:identifier type="urn:oasis:names:tc:SPML:1:0#DN">
        <spml:id>userid@system</spml:id>
        <spml:identifierAttributes>
            <spml:attr name="objectclass"><dsml:value
type="string">User</dsml:value></spml:attr>
            <spml:attr name="domain"><dsml:value
type="string">system</dsml:value></spml:attr>
            <spml:attr name="deleteResources"><dsml:value
type="string">>true</dsml:value></spml:attr>
            <spml:attr name="bulkactionid"><dsml:value
type="string">provisioning</dsml:value></spml:attr>
        </spml:identifierAttributes>
    </spml:identifier>
</spml:deleteRequest>

</s:Body>
</s:Envelope>

```

2.4 Search (using Filters)

This kind of Request type is available for the following Schemas: User, Locality, User Template, Resource, PIN.

Example of SPML Search Request for the User with the *LOGIN_NAME* "userid":

```

<?xml version="1.0" encoding="UTF-8" ?><s:Envelope
xmlns:s='http://schemas.xmlsoap.org/soap/envelope/'>

```

```

<s:Body>
<spml:searchRequest xmlns="urn:siemens:dxm:configuration:1:0"
xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"
xmlns:spml="urn:oasis:names:tc:SPML:1:0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" requestID="search-
request">
    <spml:searchBase type="urn:oasis:names:tc:SPML:1:0#DN">
        <spml:id>system</spml:id>
        <spml:identifierAttributes>
            <spml:attr name="objectclass"><dsml:value
type="string">User</dsml:value></spml:attr>
            <spml:attr name="domain"><dsml:value
type="string">system</dsml:value></spml:attr>
        </spml:identifierAttributes>
    </spml:searchBase>
    <spml:filter>
        <dsml:equalityMatch name="LOGIN_NAME"><dsml:value
type="string">userid</dsml:value></dsml:equalityMatch>
    </spml:filter>
</spml:searchRequest>
</s:Body>
</s:Envelope>

```

Paging in SPML Search Requests

The current implementation of SPML search User request allows an spml client to search for a specific user by UserID or retrieve a max of 1000 users from a domain. This is quite limited for real-life scenarios where UM is expected to manipulate thousands of users. SPML clients must have the ability to use the filtering/paging functionality offered by UM Business Logic.

Paging shall be instructed by specifying the following operation attributes in the search Request.

Param Name	Description
pageSize	The number of results returned. If set to 0 or negative, paging will be disabled
pageNumber	The requested page. 0 meaning the first page

The search Result shall also report in the operationalAttributes the following:

Param Name	Description
pageSize	the number of results returned
pageNumber	The requested page. 0 meaning the first page
numberOfPages	The total number of pages for the given request
numberOfResults	The total number of entries for the given result

If paging is not specified in the request, um spml provisioning shall consider paging with pageSize="1000" and pageNumber="0".

Sorting SPML Search Requests

Similarly, sorting shall be described in the operational attributes of the search Request. The involved attributes shall appear in the request and reply messages.

Param Name	Description
sortBy	The field to be sorted.
sortType	ASC or DESC

If the sortBy contains an invalid name or a field that is not sortable, Sorting will be discarded (and the searchReply shall not reply sorting operational attributes).

Filtering in SPML Search Requests

SPML Search Requests can incorporate DSML filters. The filter operands, as defined in DSML specification are:

Operand Name	Operand Argument	UM filter equivalent
Present	attributeName	Must be added in FilterParameters class
equalityMatch	attributeName,attributevalue	FilterOperator.EQUAL
greaterOrEqual	attributeName,attributevalue	Must be Added in FilterOperator class
lessOrEqual	attributeName,attributevalue	Must be Added in FilterOperator class
approxMatch	attributeName,attributeValue	To be decoded as : LIKE *value*
substrings	attributeName,attributeValueStart,AttributeValueEnd	To be decoded as : LIKE attributeValueStart*AttributeValueEnd
extensibleMatch	Not Applicable	Matching Rules are LDAP specific
Not	Filter	Not Available - Must be added in FilterParameters class
And	FilterSet	FilterParameters.andClause
Or	FilterSet	FilterParameters.orClause

2.5 SPML attributes

Fields	Comments	SPML Attributes for Add/Modify/Delete	DSML Attributes for Search Requests
Last Name	Standard string filter	lastName	LAST_NAME
First Name	Standard string filter	firstName	GIVEN_NAME
Middle Name	Standard string filter	middleName	MIDDLE_NAME
Company Name	Standard string filter	company	ORGANIZATION_NAME
Locality	Standard string filter	locality	LOCALITY_NAME
Notice	Standard string filter	notice	NOTICE
Bus. Phone 1	Standard string filter	businessPhone1	BUSINESS_PHONE
Bus. Phone 2	Standard string filter	businessPhone2	BUSINESS_PHONE2
Mobile Phone	Standard string filter	mobilePhone	MOBILE_PHONE
Home Phone	Standard string filter	homePhone	HOME_PHONE
Fax Number	Standard string filter	fax	FAX
E-Mail Address	Standard string filter	emailAddress	EMAIL
E-Mail Address 2	Standard string filter	alternateEmailAddress	EMAIL2
Resource *	Development shall provide the set of options		
User ID	Standard string filter	loginName	
Display Name	Standard string filter	displayName	DISPLAY_NAME
Explicit Display Name	Standard string filter	explicitDisplayName	EXPLICIT_DISPLAY_NAME
Description	Standard string filter	description	DESCRIPTION
Department	Standard string filter	department	DEPARTMENT
ONS	Standard string filter		ONSNUMBER
Pickup Group	Standard string filter	pickupGroup	CPGID
Hunt Group Id	Standard string filter	hgNotPilotId	HGIDS
Hunt Group Name	Standard string filter	hgName	HGID_NAME
Gender	Standard string filter	gender	GENDER
Language	Standard string filter	language	LANGUAGE
Home Time Zone	Exact text (ISO Format). Home Time Zone is removed from user	homeTimeZone	HOME_TIME_ZONE

	attributes in v8		
IM Address	Standard string filter	iMAddress	IM_ADDRESS
Home URL	Standard string filter	homeURL	HOME_URL
Building	Standard string filter	building	BUILDING
Room	Standard string filter	room	ROOM
Street	Standard string filter	street	STREET
Postal Code	Standard string filter	postalCode	POSTAL_CODE
City	Standard string filter	city	CITY
State/Province	Standard string filter	state_province	STATE_OR_PROVINCE
Country	Exact Text (ISO Format)	country	COUNTRY
Source	Development shall provide the set of options	sourceId	

2.6 Special cases of SPML attributes

Except for the SPML Attributes described in the above table, there are also two categories of SPML Attributes regarding the User Object that can also be provisioned through SPML. These categories are:

Custom User Attributes

This kind of attributes are configured dynamically through User Management. They consist of an Attribute ID, an Attribute Name and the type. When configuring the value of a custom user attributes for a specific user through SPML, then the attribute ID should be used.

Organizational Structure Attributes

The organizational Structure Attributes consist of an Attribute Id and an Attribute Name. They differ from the Custom User Attributes though, since an Organizational Structure cannot be added or deleted. They can only be renamed. There are seven fixed Organizational Structure Attributes. The value of an organizational Structure attribute for a specific user, can be configured through SPML by specifying its attribute Id.

Both custom user attributes and organizational structure attributes of a User are included in

the Search Response of a User Search Request only in case they have a value. Moreover, these categories can be included in the Search Request in combination with regular user Attributes (e.g. LAST_NAME).

Example of SPML Add User Request with Custom User Attribute and Organizational Structure Attributes:

```
<?xml version='1.0' encoding='UTF-8'?>
<s:Envelope xmlns:s='http://schemas.xmlsoap.org/soap/envelope/'>
<s:Body>
<spml:addRequest xmlns="urn:siemens:dxm:configuration:1:0"
xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"
xmlns:spml="urn:oasis:names:tc:SPML:1:0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" requestID="add-
request">
<spml:operationalAttributes/>
<spml:identifier type="urn:oasis:names:tc:SPML:1:0#DN">
<spml:id>userid@system</spml:id>
<spml:identifierAttributes>
<spml:attr name="objectclass"><dsml:value
type="string">User</dsml:value></spml:attr>
<spml:attr name="domain"><dsml:value
type="string">system</dsml:value></spml:attr>
<spml:attr name="bulkactionid"><dsml:value
type="string">provisioning</dsml:value></spml:attr>
</spml:identifierAttributes></spml:identifier>
<spml:attributes>
<spml:attr name="LoginName"><dsml:value
type="string">userid</dsml:value></spml:attr>
<spml:attr name="LastName"><dsml:value
type="string">lastName</dsml:value></spml:attr>
```

```

<spml:attr name="displayName"><dsml:value
type="string">displayName</dsml:value></spml:attr>

<spml:attr name="homeTimeZone"><dsml:value
type="string">America/Los_Angeles</dsml:value></spml:attr>

<spml:attr name="firstName"><dsml:value
type="string">firstName</dsml:value></spml:attr>

<spml:attr name="language"><dsml:value
type="string">en_US</dsml:value></spml:attr>

<spml:attr name="businessPhone1"><dsml:value
type="string">+3021081898043</dsml:value></spml:attr>

<spml:attr name="fax"><dsml:value
type="string">21081870013</dsml:value></spml:attr>

<spml:attr name="notice"><dsml:value
type="string">notice</dsml:value></spml:attr>

<spml:attr name="emailAddress"><dsml:value
type="string">spmluser@example.com</dsml:value></spml:attr>

<spml:attr name="customUserAttributeId"><dsml:value
type="string">0101</dsml:value></spml:attr>

<spml:attr name="orgStrAttr1"><dsml:value type="string">Company
Name</dsml:value></spml:attr>

<spml:attr name="orgStrAttr2"><dsml:value
type="string">Department</dsml:value></spml:attr>

<spml:attr name="orgStrAttr5"><dsml:value
type="string">Office</dsml:value></spml:attr>

</spml:attributes>

</spml:addRequest>

</s:Body>

</s:Envelope>

```

In the following example of SPML Search Request with Custom User Attributes, Organizational Structure Attributes and regular attributes we are searching for the users whose *orgStrAttr2* is like "Company", the *customUserAttributeId1* is like "0101", the

customUserAttributeId2 is "7676" and the *LAST_NAME* is "userlastname":

```
<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <spml:searchRequest xmlns="urn:siemens:dxm:configuration:1:0"
xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"
xmlns:spml="urn:oasis:names:tc:SPML:1:0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" requestID="search-
request">
      <spml:searchBase type="urn:oasis:names:tc:SPML:1:0#DN">
        <spml:id>system</spml:id>
        <spml:identifierAttributes>
          <spml:attr name="objectclass">
            <dsml:value type="string">User</dsml:value>
          </spml:attr>
          <spml:attr name="domain">
            <dsml:value type="string">system</dsml:value>
          </spml:attr>
        </spml:identifierAttributes>
      </spml:searchBase>
      <spml:filter>
        <dsml:and>
          <dsml:approxMatch name=" orgStrAttr2"><dsml:value
type="string">Company</dsml:value></dsml:approxMatch>
          <dsml:approxMatch name="customUserAttributeId1"><dsml:value
type="string">0101</dsml:value></dsml:approxMatch>
          <dsml:equalityMatch name="--
customUserAttributeId2"><dsml:value
```

```
type="string">7676</dsml:value></dsml:equalityMatch>
    <dsml:equalityMatch name="LAST_NAME"><dsml:value
type="string">userlastname</dsml:value></dsml:equalityMatch>
    </dsml:and>
</spml:filter>
</spml:searchRequest>
</s:Body>
</s:Envelope>
```

2 Design Aspects

User Templates

User Management Provisioning Engine is based on the Concept of PSO (Provisioning Service Objects). This means that incoming requests are pre- processed and converted to common internal method calls. In essence this means that every provisioning operation that is supported by one method (e.g. LDAP) is by default available to the other Methods (CSV,SPML). This means that development impacts are expected to be trivial (e.g. for CSV import it should be only necessary to allow the User Template column to pass validation).

Filtering for SPML

SPML filters are supported (parsed & passed to the UmUserPSO.search method). Paging and sorting must also be provided in a way consistent to the UC Provisioning. (passed as operational attributes of the Search Request).

Authentication Against the UM Server

SPML Servlet of the CMP/UM is located under /symphonia-umspmlresponder/services/SpmlSoapService. BASIC HTTP Authentication is supported.

The following snippet shows how it is done using Apache HttpClient v3.0.1

```
URI uri = new URI("https://" + host + ":" + port+ "/symphonia-umspmlresponder/services/SpmlSoapService"); HttpClient client = new HttpClient(); client.getParams().setAuthenticationPreemptive(true); client.getState().setCredentials(new AuthScope(host, port),new UsernamePasswordCredentials(adminUser, password));
```

SPML request are SOAP messages that are submitted to the server via POST Methods

```
String head = "<?xml version='1.0' encoding='UTF-8'?>\n" + "<s:Envelope
xmlns:s='http://schemas.xmlsoap.org/soap/envelope/'>\n" +
"<s:Body>\n"; String body="" ; //The actual SPML Request . See sections
below String tail = "</s:Body>\n" + "</s:Envelope>\n"; PostMethod method
= new PostMethod(uri.toString()); method.setRequestHeader("SOAPAction",
"searchRequest"); //or addRequest etc. RequestEntity re
= new StringRequestEntity(head + body + tail); method.setRequestEntity(re);
```

3 Request Templates

CMP/UM uses the SPML parsers provided by Unify (c) DirX Metadirectory. These support the SPML v1 standard

(<http://www.oasis-open.org/committees/download.php/4137/os-pstc-spml-core-1.0.pdf>)

Attribute sets are enclosed in <attribute> tags

```
<spml:attributes> <spml:attr name='objectclass'> <dsml:value
type='string'>User</dsml:value> </spml:attr> <spml:attr name='domain'> <dsml:value
type='string'>System</dsml:value> </spml:attr> </spml:attributes>
```

Attributes of the Identifier Class are grouped under the IdentifierAttributes tag

```
<spml:searchBase type='urn:oasis:names:tc:SPML:1:0#DN'> <spml:id>user1@system</spml:i
d> <spml:identifierAttributes> <spml:attr name='objectclass'> <dsml:value
type='string'>User</dsml:value> </spml:attr> <spml:attr name='domain'> <dsml:value
type='string'>system</dsml:value> </spml:attr> </spml:identifierAttributes>
</spml:searchBase>
```

SchemaRequest

```
<schemaRequest xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"
xmlns="urn:oasis:names:tc:SPML:1:0" requestID="SchemaRequest-1"
execution="urn:oasis:names:tc:SPML:1:0#synchronous"> <operationalAttributes> <attr na
```

```

me="OperationalAttribute"> <dsml:value type="string">OA1</dsml:value> <dsml:value
type="string">OA2</dsml:value> </attr>
</operationalAttributes> <providerIdentifier providerIDType="urn:oasis:names:tc:SPML:
1:0#URN"> <providerID>symphonia-
umspmlresponder</providerID> </providerIdentifier> <schemaIdentifier schemaIDType="ur
n:oasis:names:tc:SPML:1:0#GenericString"> <schemaID>User</schemaID>
</schemaIdentifier> </schemaRequest>

```

SearchRequest

```

<searchRequest xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"
xmlns="urn:oasis:names:tc:SPML:1:0" requestID="searchRequest-
ID"> <operationalAttributes> <attr name="PageNumber"> <dsml:value
type="string">1</dsml:value> </attr> <attr name="PageSize"> <dsml:value
type="string">1000</dsml:value> </attr>
</operationalAttributes> <searchBase type="urn:oasis:names:tc:SPML:1:0#DN"> <id>user1
@system</id> <identifierAttributes> <attr name="ObjectClass"> <dsml:value
type="string">User</dsml:value> </attr> <attr name="domain"> <dsml:value
type="string">system</dsml:value> </attr> </identifierAttributes> </searchBase> <!--
For supported filtering options see sectio above --> <filter> <dsml:and>
<dsml:equalityMatch name="firstName"> <dsml:value type="string">John</dsml:value>
</dsml:equalityMatch> </dsml:and>
</filter>
<!-- Support for retrieving subsets of attributes is not supported and out of the
Scope of FRN6030 -->
<attributes> <dsml:attribute name="AttributeName"/> </attributes> </searchRequest>

```

Retrieving subsets of attributes does not provide any performance gains and therefore it is not supported.

AddRequest

```

<spml:addRequest xmlns="urn:siemens:dxm:configuration:1:0"
xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"
xmlns:spml="urn:oasis:names:tc:SPML:1:0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" requestID="add-request"> <spml:operationalAttributes

```

```

/> <spml:identifier type="urn:oasis:names:tc:SPML:1:0#DN"> <spml:id>test@system</spml
:id> <spml:identifierAttributes> <spml:attr name="ObjectClass"> <dsml:value
type="string">User</dsml:value> </spml:attr> <spml:attr name="domain"> <dsml:value
type="string">system</dsml:value> </spml:attr> </spml:identifierAttributes>
</spml:identifier>
<spml:attributes> <spml:attr name="AttributeName"> <dsml:value type="string">Attribut
eDescription</dsml:value> </spml:attr> </spml:attributes> </spml:addRequest>

```

ModifyRequest

```

<spml:modifyRequest xmlns="urn:siemens:dxm:configuration:1:0"
xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"
xmlns:spml="urn:oasis:names:tc:SPML:1:0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" requestID="modify-request"> <spml:operationalAttributes
/> <spml:identifier type="urn:oasis:names:tc:SPML:1:0#DN"> <spml:id>test@system</spml
:id> <spml:identifierAttributes> <spml:attr name="ObjectClass"> <dsml:value
type="string">User</dsml:value> </spml:attr> <spml:attr name="domain"> <dsml:value
type="string">system</dsml:value> </spml:attr> </spml:identifierAttributes>
</spml:identifier> <spml:modifications> <spml:modification name="firstName"
operation="replace"> <dsml:value type="string">Marios</dsml:value>
</spml:modification> </spml:modifications> </spml:modifyRequest>

```

SPML Modification also supports operations **Add** and **Delete** but these are irrelevant to the current SOA Architecture and are not supported.

DeleteRequest

```

<deleteRequest xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"
xmlns="urn:oasis:names:tc:SPML:1:0" requestID="delete-
request"> <operationalAttributes
/> <identifier type="urn:oasis:names:tc:SPML:1:0#DN"> <id>test@system</id> <identifie
rAttributes> <attr name="ObjectClass"> <dsml:value type="string">User</dsml:value>
</attr> <attr name="domain"> <dsml:value type="string">system</dsml:value> </attr>
</identifierAttributes> </identifier> </deleteRequest>

```

