SAP NetWeaver® Identity Management
Identity Services

Architectural overview

Version 7.2 Rev 6
Preface

The product

The SAP NetWeaver Identity Management Identity Services provides web services access to identity information stored in an identity store in the Identity Center or some other application that can be accessed from the Virtual Directory Server. The Identity Services accepts SPML (Service Provisioning Markup Language) requests. It acts as an abstraction layer between identity services clients and the identity information. This reduces the complexity of access as much of the connection details (location, protocols, repository types and so on) are hidden from the clients.

The reader

This manual is intended for people who want an overview of the Identity Services.

Prerequisites

To get the most benefit from this manual, you should have the following knowledge:

- Knowledge of the Identity Center.
- Knowledge of the Virtual Directory Server.
- Knowledge of deploying web services on SAP NetWeaver.

The following software is required:

- SAP NetWeaver Identity Management Virtual Directory Server 7.2 SP9 or newer, correctly installed and licensed.
- SAP NetWeaver Identity Management Identity Center 7.2 SP9 or newer, correctly installed and licensed.
- One of the following SAP NetWeaver versions:
  - SAP NetWeaver AS Java as of Release 7.0 SP14 or higher (SAP NetWeaver 7.0).
  - SAP NetWeaver Composition Environment 7.1 Including Enhancement Package 1 (SAP NetWeaver CE 7.1 EHP 1).
  - SAP NetWeaver Composition Environment 7.2 (SAP NetWeaver CE 7.2).
  - SAP NetWeaver 7.3.
  - SAP NetWeaver 7.3 Including Enhancement Package 1 (SAP NetWeaver 7.3 EHP 1).
  - SAP NetWeaver 7.4.

The manual

This document gives an overview of the architecture of Identity Services. The document also describes the operations that can be submitted to Identity Services.
Related documents

You can find useful information in the following documents:

- The tutorials for the Identity Center
- The tutorials for the Virtual Directory Server
- Information about SPML, [http://www.oasis-open.org/specs/index.php#spmlv1.0](http://www.oasis-open.org/specs/index.php#spmlv1.0)
- *SAP NetWeaver Identity Management Compliant provisioning using SAP Access Control – Architectural overview.*
- *SAP NetWeaver Identity Management Compliant provisioning using SAP Access Control - Configuration guide.*
- For more documentation for SAP NetWeaver, see Help Portal ([http://help.sap.com](http://help.sap.com)).
# Table of contents

**Introduction** ........................................................................................................................... 1  
   The role of the Identity Center ................................................................................................. 2  
   Requirements for an Identity Services solution ......................................................................... 2  
   Identity Services ...................................................................................................................... 3  
   Dealing with a heterogeneous environment ............................................................................ 4  

**Objects and information levels in the Identity Services** ...................................................... 5  
   Entry types in the identity store ............................................................................................... 5  
   The virtual tree ....................................................................................................................... 5  
   Information levels .................................................................................................................. 6  

**SPML identifiers** .................................................................................................................. 8  
   Simplifications ....................................................................................................................... 8  
   Accepted identifier formats .................................................................................................... 9  
   Special identifiers ................................................................................................................ 10  
   Identifier processing ............................................................................................................ 10  

**Operation overview** ............................................................................................................ 11  
   Sequence of operations ......................................................................................................... 11  
   The example structure ......................................................................................................... 11  

**Special operations** .............................................................................................................. 12  
   General limitation for mskey as filtering attribute ................................................................. 13  
   Retrieving overview information (info) ................................................................................ 13  
   Listing available systems (listsystems) .................................................................................. 16  
   Listing available applications (listapplications) ................................................................. 18  
   Listing available privileges (listprivileges) ......................................................................... 20  
   Listing available roles (listroles) ........................................................................................ 24  
   Listing users (listusers) ...................................................................................................... 28  
   Retrieving the schema (schema) .......................................................................................... 32  
   Obtaining the result of a provisioning operation (requeststatus/auditlog) ........................ 36  
   Obtaining the result of a role assignment (roleassignmentresult) ...................................... 38  
   Listing multiple entries (list) ............................................................................................... 39  

**Provisioning operations** ..................................................................................................... 40  
   Adding a person entry ....................................................................................................... 40  
   Modifying a person entry ................................................................................................. 42  
   Deleting a person entry .................................................................................................... 43  

**Search operations** ............................................................................................................... 44  

© Copyright 2014 SAP AG. All rights reserved.
Introduction

An SAP system landscape at a customer normally consists of a large number of applications. Many of these applications are still "islands" with regards to identity information, leading to complex management, security risks, and lack of an overview of who is allowed to do what, which is required by regulations.

The identity management hub is needed to accomplish the following:

- Enable efficient and secure management of identities.
- Complement the existing security functionality (single sign-on, digital signature, authorization management, encryption, and web services security).
- Become the authoritative source of identity information in the whole SAP environment. As a result, the number of applications that needs access to it is considerable.

The following illustration shows the SAP NetWeaver Identity Management's role in this scenario.
The role of the Identity Center

One of the components of Identity Management is the Identity Center. Within the Identity Center, the identity store contains the identity data of all the applications where identity information is managed. This identity store is then used to provision the users and access rights to the applications, which can be controlled by role assignments or by rules.

When the Identity Center is an active part and updates or retrieves data in other applications’ stores, the following issues may arise:

- **Security**
  For security and policy reasons, the access to the application should be limited and controlled.

- **Control**
  Applications need to read and update data in the identity store, in their own pace, when they need it.

To solve this, there is a need for a solution that will give external applications access to the identity data and make it possible to write data to the identity store, on their own premises. The role of the Identity Services is to provide an access point for other applications to read and update identity data. Provisioning to external systems is done using a properly configured task structure, possibly based on a provisioning framework.

Requirements for an Identity Services solution

There are several requirements that such a solution has to fulfill:

- **Standard protocols**
  In order to make the service available to variety of applications, it has to be based on standard protocols (like SPML and LDAP). Since access protocols evolve (and new protocols gain popularity), it has to be possible to extend the available protocol set.

- **Flexible architecture**
  The properties and requirements of the Identity Services itself may change over time. The solution must be capable of coping with such changes in a way that it removes the burden of changes from Identity Services consumers.

- **Secure**
  The number of potential Identity Services consumers is large. Each consumer may have different needs and requirements. Hence, it is crucial that the solution has the means of controlling access to the identity data.
Identity Services

The Identity Services solution is based on SAP NetWeaver Identity Management Virtual Directory Server, which is (out-of-the-box) capable of satisfying the majority of the requirements listed previously.

The Virtual Directory Server offers native support for multiple inbound protocols (as well as extensible framework for adding additional inbound protocols).

The architecture of the solution is shown below.

In the current version of the Identity Services, SPML and LDAP are supported. This means that applications can use any of these protocols to communicate with solution.
Dealing with a heterogeneous environment

A heterogeneous environment typically consists of identity data in multiple external applications:

- Identity stores in the Identity Center: The provisioning solution in each of those identity stores is able to manage multiple SAP and non-SAP systems.
- Other SAP applications that are not part of an Identity Management provisioning solution.
- Other non-SAP applications that are not part of an Identity Management provisioning solution.

The Virtual Directory Server implements a structure called a virtual directory tree. It is a structure that organizes all managed applications so that each of them can be addressed through a unique identifier. A unique identifier, in this context, corresponds to a distinguished name in the virtual directory tree, but is mapped to a unique identifier within the application.

In addition, the Virtual Directory Server has built-in connectors (and an extensible connector framework) for a variety of the applications. Most important, the Virtual Directory Server has a connector for SAP NetWeaver Identity Management Identity Center, so it can execute operations directly in identity store.

The Virtual Directory Server provides a range of additional services as virtualization, namespace conversion, attribute and schema mapping, attribute value modification etc. These services may be crucial for resolving requirements regarding the flexibility of the Identity Services solution, as mentioned previously.

The current version of the Identity Services is limited to enabling identity stores in the Identity Center for Identity Services.
Objects and information levels in the Identity Services

The main purpose of the Identity Services is to provide web services access to identity data in the Identity Center. Through web services, external clients are able to manage users and privileges (technical roles) and roles (business roles). The Identity Services accepts operations to create and modify users, as well as assigning and removing privilege assignments.

In addition, there are a number of operations for retrieving information about the configuration and the system.

Normally, one Identity Services configuration will be connected to only one identity store, but it is possible to configure multiple.

Entry types in the identity store

There are normally several entry types within an identity store. The Identity Services uses mostly the following three entry types when performing its operations:

- **MX_PERSON**
  The entry type for persons in the identity store.

- **MX_PRIVILEGE**
  The entry type for privileges (technical roles) that can be used to control provisioning in the connected external systems/applications. The privileges are stored as a flat structure.

- **MX_ROLE**
  The entry type for roles (business roles) that can be used to control provisioning in the connected external systems/applications.

The virtual tree

When configuring an Identity Services solution in the Virtual Directory Server, you define a virtual tree. Below the root, the first level is always the identity store. If only one identity store is used, you can use the starting point to avoid having to address this level. Multiple identity stores may be in the same or in different Identity Centers.
Information levels

There are five information levels in the Identity Services:

- The system level
- The application level
- The privilege (technical role) level
- The role (business role) level
- The user level

The system level

The system information level corresponds to the identity stores in the Identity Center.

A typical Identity Center installation has only one identity store, but the Identity Services is designed to work with multiple identity stores.

The most important property of an entity on system level is its naming context. This is the distinguished name suffix that is used to uniquely address each particular system in the Identity Services solution.

Normally, the naming context must be a part of the all received SPML requests' identifiers – otherwise, the Identity Services is not able to determine the destination of the requested SPML operation (that is, to select the application). In some special cases, the naming context is not required. See page 8.

The Identity Services implements a special SPML operation that lists all managed systems and their properties. See section Listing available systems (listsystems) on page 16 for more information about operation listsystems.

The application level

The application information level gives a possibility to group the privileges in the identity store. For instance, the provisioning solutions implemented in the Identity Centers can normally manage (carry out provisioning/de-provisioning operations) to multiple SAP and non-SAP systems. Then each of these systems can be defined as an application, and the privileges that control provisioning/deprovisioning within this system can be connected to this application (by using the privilege attributes MX_APPLICATION_ID and/or MX_REPOSITORYNAME).
The Identity Services offers a special SPML operation that lists all managed applications for each of the managed systems. See section *Listing available applications (listapplications)* on page 18 for more information about the operation *listapplications*.

**The privilege level**

This information level consists of the privileges (technical roles) that may be assigned to entries in the identity store. The privileges are represented by entries with the entry type MX_PRIVILEGE in the identity store.

The privileges may contain the information about the applications (referred to by attributes MX_APPLICATION_ID and/or MX_REPOSITORYNAME).

The Identity Services implements a special SPML operation that lists privileges for all managed systems and managed applications. See section *Listing available privileges (listprivileges)* on page 20 for more information about the operation *listprivileges*.

**The role level**

This information level consists of the roles (business roles) that may be assigned to entries in the identity store. The roles are represented by entries with the entry type MX_ROLE in the identity store.

The Identity Services implements a special SPML operation that lists roles for all managed systems. See section *Listing available roles (listroles)* on page 24 for more information about the operation *listroles*.

**The user level**

This information level consists of the user entries in the identity store, represented by the entry type MX_PERSON in the identity store.

The Identity Services implements a special SPML operation that lists users for all managed systems. See section *Listing users (listusers)* on page 28 for more information about the operation *listusers*.
SPML identifiers

The Identity Services accepts and processes SPML requests with valid SPML identifiers. As the Identity Services can manage multiple identity stores, all incoming SPML identifiers must contain information about which of the supported systems the operation is supposed to be carried out in.

A valid SPML identifier consists of two different parts:

- **A unique part**
  Unique description of the object.

- **A naming context**
  The position of the object in the given structure.

In the SPML identifier:

```
CN=John Parrot,OU=people,O=myorg
```

*CN=John Parrot* is the unique part of the identifier, while *OU=people,O=myorg* is the suffix and tells where the object (given by the unique part) is placed in the overall naming context.

**Simplifications**

To simplify the work with the solution, Identity Services accepts identifiers that do not fully comply with the rules of a valid SPML identifier. In that case, the Identity Services applies certain rules and constructs a valid SPML identifier that will be used for operation execution.

The following sections describe the valid simplifications and the rules that are used to construct valid SPML identifiers.

**Default system**

In each Identity Services configuration, one of the systems is considered the default system.

The Identity Services accepts identifiers that lack the suffix part of the SPML identifier. In that case the Identity Services solution will construct a full a SPML identifier using the naming context of the default system.

**Short unique part**

The Identity Services may accept simple identifiers that are not in DN format.

In that case, the Identity Services solution will construct the unique part of the identifier by adding "CN=" to it, and then apply the rules for adding a naming context.

**Combination**

It is possible to supply a short unique part and still select the naming context that is not default. In that case, the identifiers are on the form:

```
<short unique identifier part>@<selected system's id>
```
# Accepted identifier formats

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;unique part&gt;,&lt;naming context part&gt;</code></td>
<td>This is the standard operation mode. The identifier is given on DN form. Identity Services will not post process it, but use it as it is.</td>
</tr>
<tr>
<td>Example:</td>
<td>CN=john parrot,OU=people,O=ids1</td>
</tr>
<tr>
<td><code>&lt;short unique part&gt;</code></td>
<td>Since the &quot;CN=&quot; part is missing, the Identity Services assumes that this is the &quot;short unique part mode&quot;. It will first add &quot;CN=&quot; and construct the unique part in the DN form (CN=john parrot). Then, it will append the naming context for the default system (here, we assume it is system with id &quot;defsys&quot; and naming context &quot;OU=people,O=ids1&quot;).</td>
</tr>
<tr>
<td>Example:</td>
<td>john parrot</td>
</tr>
<tr>
<td><code>&lt;short unique part&gt;@&lt;naming context&gt;</code></td>
<td>Since the &quot;CN=&quot; part is missing, the Identity Services assumes that this is the &quot;short unique part mode&quot;. It will first add &quot;CN=&quot; and construct the unique part on the DN form (CN=john parrot). Then, since the ‘@’ is found (i.e. the consumer explicitly specified a system), it will append the specified system's naming context to construct the full SPML identifier.</td>
</tr>
<tr>
<td>Example:</td>
<td>john parrot@anothersys</td>
</tr>
<tr>
<td><code>&lt;unique part&gt;</code></td>
<td>Note: This is an example of an illegal identifier. Since &quot;CN=&quot; is found, the Identity Services will assume &quot;full SPML identifier mode&quot; and not carry out any preprocessing of the received identifier.</td>
</tr>
<tr>
<td>Example:</td>
<td>CN=john parrot</td>
</tr>
</tbody>
</table>

The result is an invalid identifier.
Special identifiers

The Identity Services defines a set of special identifiers that may be used to perform specific operations. Typically, these operations are used to obtain meta-information about the Identity Services and construct valid SPML identifiers for their SPML requests.

For example, to use Identity Services, a potential consumer needs the information about managed systems, managed applications and available privileges. This information is obtained using special operations.

The list of special identifiers and description of the functionality of each operation is described on page 12.

Identifier processing

The Identity Services system intercepts all incoming SPML requests and checks the received identifier in the following manner:

1. It verifies if the received identifier contains a "@". If it does, the identifiers are split in the unique parts and the suffix part. The context part is stored for later use (as "Selected" context).
   If no "@" is present, the "Selected" context is set to the default context.

2. Next, it checks if the unique part of the identifier belongs to the list of known special operations. If it does, the operation is pre-processed (if needed) and the new identifier is created using the "Selected" context.

3. If the identifier is not in the list of special operation, the system checks if this is an identifier on DN form.
   If it is on DN form, the request is sent to the Virtual Directory Server without modification. The system assumes that the naming context is filled in by the consumer.
   If the identifier does not contain a DN, the system creates a new identifier using the "Selected" naming context and submits the request to the Virtual Directory Server.
Operation overview

There are different types of operations you can submit to the Identity Services:

<table>
<thead>
<tr>
<th>Operation type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Special operations</td>
<td>These operations are used to discover the systems, applications and privileges of the system. There is also an operation that can be used to view the result of a provisioning operation.</td>
</tr>
<tr>
<td>Provisioning operations</td>
<td>These are standard SPML operations Add, Modify and Delete.</td>
</tr>
<tr>
<td>Search operations</td>
<td>These operations are used to listing and viewing information about entries.</td>
</tr>
</tbody>
</table>

Sequence of operations

A typical sequence to perform these operations will be:

1. Perform the special operations necessary to get an overview of the system you access with the Identity Services.
2. Perform provisioning operations.
3. Perform the special operation to see the status of the provisioning operation.
4. Perform search operations on the modified data.

The example structure

In the following descriptions, the following structure is used in the examples of the operations:
Special operations

The following chapter describes the set of special operations that are supported by the Identity Services.

There may be several ways of executing the same operation. Typically, multiple identifiers may lead to the same operation. The special operations are provided from convenience. All these operations can be executed using the normal search operations using valid SPML identifiers.

This table gives an overview of the operations. The details are described in the following sections.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrieve overview information</td>
<td>Returns overview information about the system. Among other attributes it returns:</td>
</tr>
<tr>
<td></td>
<td>- the list of the other special operations</td>
</tr>
<tr>
<td></td>
<td>- the list of known naming contexts</td>
</tr>
<tr>
<td></td>
<td>- the default naming context</td>
</tr>
<tr>
<td>List available systems</td>
<td>Lists all available systems and their basic properties.</td>
</tr>
<tr>
<td>List available applications</td>
<td>Lists all available applications for a particular system.</td>
</tr>
<tr>
<td>List available privileges</td>
<td>Lists all available privileges for particular system. It also can list</td>
</tr>
<tr>
<td></td>
<td>privileges for particular applications. A special case of the general</td>
</tr>
<tr>
<td></td>
<td>operation <strong>list</strong></td>
</tr>
<tr>
<td>List available persons</td>
<td>Lists all available person entries for a specific system. A special</td>
</tr>
<tr>
<td></td>
<td>case of the general operation <strong>list</strong></td>
</tr>
<tr>
<td>List available roles</td>
<td>Lists all available roles for a specific system or application. A special</td>
</tr>
<tr>
<td></td>
<td>case of the general operation <strong>list</strong></td>
</tr>
<tr>
<td>Retrieve schema</td>
<td>Returns the existing entry types and attributes.</td>
</tr>
<tr>
<td></td>
<td>Alternatively, it is possible to obtain schema by sending a valid</td>
</tr>
<tr>
<td></td>
<td>getSchema SPML request.</td>
</tr>
<tr>
<td>Retrieve provisioning operation</td>
<td>Both return information about the state of a specific provisioning</td>
</tr>
<tr>
<td>status</td>
<td>operation. The operation <strong>requeststatus</strong> is preferred to obtain the</td>
</tr>
<tr>
<td></td>
<td>status of the provisioning operation. The operation <strong>auditlog</strong> is</td>
</tr>
<tr>
<td></td>
<td>obsolete.</td>
</tr>
<tr>
<td>Listing multiple entries</td>
<td>A search operation that returns multiple entries. Using the filters, the</td>
</tr>
<tr>
<td></td>
<td>general operation list can be used for retrieving information</td>
</tr>
<tr>
<td></td>
<td>about different entries. E.g. to list the privileges with the general</td>
</tr>
<tr>
<td></td>
<td>operation= <strong>list</strong>, the filter needs to be</td>
</tr>
<tr>
<td></td>
<td>(objectclass=MX_PRIVILEGE).</td>
</tr>
<tr>
<td></td>
<td>Operations <strong>listprivileges</strong>, <strong>listusers</strong> and <strong>listroles</strong> are special</td>
</tr>
<tr>
<td></td>
<td>cases of the general <strong>list</strong> operation.</td>
</tr>
<tr>
<td>Retrieve role assignment status</td>
<td>Operation returning the result of the role assignment.</td>
</tr>
<tr>
<td>operation</td>
<td></td>
</tr>
</tbody>
</table>
### General limitation for mskey as filtering attribute

In general, mskey is not searchable even if returned, and thus not possible to use to filter the information provided by the operations. The reason for this is that mskey is an internal attribute.

### Retrieving overview information (info)

The info operation corresponds to the RootDSE operation for LDAP directories. It returns overview information about the Identity Services system.

The most important information this operation returns is the list of all special operations that consumer may execute on Identity Services system.

### Operation properties

<table>
<thead>
<tr>
<th>Key</th>
<th>Value/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation</td>
<td>SPML Search operation</td>
</tr>
<tr>
<td>Starting point</td>
<td>operation=info or &lt;empty string&gt;</td>
</tr>
<tr>
<td>Search type</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Attributes requested</td>
<td>Not applicable, all info attributes returned</td>
</tr>
<tr>
<td>Filter</td>
<td>Not applicable</td>
</tr>
</tbody>
</table>

### Operation results

A single entry is returned. The identifier of the returned entry is equal to the identifier of the original operation.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Value/Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>supportedspmlversion</td>
<td>The SPML value supported. Currently SPML v 1.</td>
<td>SPMLv1</td>
</tr>
<tr>
<td>supportedsystemtypes</td>
<td>The current version of Identity Services supports only SAP NetWeaver Identity Management.</td>
<td>Currently: SAPNWIdM</td>
</tr>
<tr>
<td>namingcontexts</td>
<td>A semi-colon separated list of DN suffixes for each of the available systems.</td>
<td>Example: ou=idm1,o=ids; ou=idm2,o=ids</td>
</tr>
<tr>
<td>defaultnamingcontext</td>
<td>The default naming context. If the system receives an operation that does not have namingcontexts as part of the identifier, this one will be added to the received identifier.</td>
<td>ou=idm1,o=ids</td>
</tr>
<tr>
<td>Attribute</td>
<td>Description</td>
<td>Value/Example</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>defaultsystemid</td>
<td>The ID of the default system. This is the system whose naming context (see defaultnamingcontext) will be appended to identifier if a system receives an operation without naming context. This is one of the properties of the node pointed to by defaultnamingcontext.</td>
<td>idm1</td>
</tr>
<tr>
<td>operationset</td>
<td>A comma separated list of the identifiers that the system may accept in order to execute special operations. Operation auditlog, although available, is obsolete.</td>
<td>operation=info, operation=schema, operation=list, (operation=auditlog), operation=listsystems, operation=listapplications, operation=listprivileges, operation=listroles, operation=listusers, operation=requeststatus, operation=roleassignmentresult, operation=roledefinitionresult</td>
</tr>
<tr>
<td>topsuffix</td>
<td>The DN of the top node in the virtual tree.</td>
<td>o=ids</td>
</tr>
<tr>
<td>defaultrdnkey</td>
<td>The relative distinguished name part for this particular system.</td>
<td>cn</td>
</tr>
</tbody>
</table>

**Example**

**SPML request**

```xml
<SOAP-ENV:Body>
        <spml:searchBase type="urn:oasis:names:tc:SPML:1:0#GUID">
            <spml:id>operation=info</spml:id>
        </spml:searchBase>
        <dsml:filter>
            <dsml:present name="objectclass"></dsml:present>
        </dsml:filter>
    </spml:searchRequest>
</SOAP-ENV:Body>
```
SPML response

```xml
<SOAP-ENV:Body>
  <spml:searchResponse result="urn:oasis:names:tc:SPML:1:0#success"
    xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"
    xmlns:spml="urn:oasis:names:tc:SPML:1:0">
    <searchResultEntry>
      <spml:identifier>
        <spml:id>operation=info</spml:id>
      </spml:identifier>
      <spml:attributes>
        <dsml:attr name="supportedsystemtypes">
          <dsml:value type="xsd:string">[SAPNWIdM]</dsml:value>
        </dsml:attr>
        <dsml:attr name="operationset">
          <dsml:value type="xsd:string"/>
        </dsml:attr>
        <dsml:attr name="defaultnamingcontext">
          <dsml:value type="xsd:string">ou=nwidm1,o=ids</dsml:value>
        </dsml:attr>
        <dsml:attr name="supportedspmlversion">
          <dsml:value type="xsd:string">SPMLv1</dsml:value>
        </dsml:attr>
        <dsml:attr name="topsuffix">
          <dsml:value type="xsd:string">o=ids</dsml:value>
        </dsml:attr>
        <dsml:attr name="namingcontexts">
          <dsml:value type="xsd:string">ou=nwidm1,o=ids</dsml:value>
        </dsml:attr>
        <dsml:attr name="defaultrdnkey">
          <dsml:value type="xsd:string">cn</dsml:value>
        </dsml:attr>
        <dsml:attr name="defaultsystemid">
          <dsml:value type="xsd:string">idm1</dsml:value>
        </dsml:attr>
      </spml:attributes>
    </searchResultEntry>
  </spml:searchResponse>
</SOAP-ENV:Body>
```
Listing available systems (listsystems)

After having retrieved the overview information with the previous operation, this is the most natural second operation that the consumer may execute on Identity Services solution. It lists all available systems that Identity Services can expose.

There are multiple ways of accessing this information.

Operation properties

<table>
<thead>
<tr>
<th>Key</th>
<th>Value/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation</td>
<td>SPML Search operation</td>
</tr>
<tr>
<td>Starting point</td>
<td>operation=listsystems</td>
</tr>
<tr>
<td>Search type</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Attributes requested</td>
<td>* or any attribute subset (see available attributes below in section Returned entries). Note that namingcontext is always appended, regardless if it is requested or not.</td>
</tr>
<tr>
<td>Filter</td>
<td>Complex or simple filter (exact match, substring and present filter) involving any of the available attributes (except for naming context) is allowed.</td>
</tr>
</tbody>
</table>

Returned entries

This operation returns a list of entries.

The identifier for each of the returned entries is the naming context of the system in question (the same value is also returned in the attribute namingcontext, see below)

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Value/Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>objectclass</td>
<td>In the current version, only the IDS_SYSTEM object class is used.</td>
<td>IDS_SYSTEM</td>
</tr>
<tr>
<td>systemid</td>
<td>The unique name of the system. It may later be used in listapplications and/or other operations where filtering on the system is possible.</td>
<td>idm1</td>
</tr>
<tr>
<td>systemtype</td>
<td>The type of system.</td>
<td>In the current version: SAPNWIdM</td>
</tr>
<tr>
<td>description</td>
<td>Description of the system (optional).</td>
<td>description</td>
</tr>
<tr>
<td>namingcontext</td>
<td>The DN suffix needed to address this particular system. It may later be used in all operations to specify the desired system for operation.</td>
<td>ou=idm1,o=ids</td>
</tr>
<tr>
<td>rdnkey</td>
<td>The relative distinguished name part for this particular system.</td>
<td>cn</td>
</tr>
</tbody>
</table>
Example

SPML request

```xml
<SOAP-ENV:Body>
  <spml:searchRequest xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"
   xmlns:spml="urn:oasis:names:tc:SPML:1:0">
    <spml:searchBase type="urn:oasis:names:tc:SPML:1:0#GUID">
      <spml:id>operation=listsystems</spml:id>
    </spml:searchBase>
    <dsml:filter>
      <dsml:present name="objectclass"></dsml:present>
    </dsml:filter>
  </spml:searchRequest>
</SOAP-ENV:Body>
```

SPML response

```xml
<SOAP-ENV:Body>
   xmlns:spml="urn:oasis:names:tc:SPML:1:0">
    <searchResultEntry>
      <spml:identifier>
        <spml:id>ou=nwidm1,o=ids</spml:id>
      </spml:identifier>
      <spml:attributes>
        <dsml:attr name="objectclass">
          <dsml:value type="xsd:string">IDS_SYSTEM</dsml:value>
        </dsml:attr>
        <dsml:attr name="namingcontext">
          <dsml:value type="xsd:string">ou=nwidm1,o=ids</dsml:value>
        </dsml:attr>
        <dsml:attr name="description">
          <dsml:value type="xsd:string">This is a description for idm1</dsml:value>
        </dsml:attr>
        <dsml:attr name="systemtype">
          <dsml:value type="xsd:string">SAPNWIdM</dsml:value>
        </dsml:attr>
        <dsml:attr name="rdnkey">
          <dsml:value type="xsd:string">cn</dsml:value>
        </dsml:attr>
        <dsml:attr name="systemid">
          <dsml:value type="xsd:string">idm1</dsml:value>
        </dsml:attr>
      </spml:attributes>
    </searchResultEntry>
  </spml:searchResponse>
</SOAP-ENV:Body>
```
Listing available applications (listapplications)

Each Identity Center (the system level) can manage multiple back-end applications. The Identity Services consumer can get information about available applications using following commands.

Operation properties

<table>
<thead>
<tr>
<th>Key</th>
<th>Value/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation</td>
<td>SPML Search operation</td>
</tr>
<tr>
<td>Starting point</td>
<td>operation=listapplications</td>
</tr>
<tr>
<td>Search type</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Attributes requested</td>
<td>* or any attribute subset (see available attributes for this operation in section Returned data on page 18)</td>
</tr>
<tr>
<td>Filter</td>
<td>(objectclass=ids_application) and/or filtering involving applicationid (exact match only) are allowed. See the section about filtering per system below.</td>
</tr>
</tbody>
</table>

The rules about specifying a specific system for the operation can be applied here (see section Filtering applications per system below).

Filtering applications per system

Supplying an SPML identifier that consists only of:

```
listapplications
```

will list all applications managed by the default system.

In order to list applications that are managed by a specific system you should create an identifier by specifying the relevant system ID in the following way:

```
listapplications@<relevant system id>
```

Returned data

A list of entries is returned. The identifiers of the returned entries are on the form:

```
cn = <application ID>, <the System Naming Context>
```

where the system naming context is the one that was selected by executed listapplications command.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Value/Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>objectclass</td>
<td>Entry type of the object</td>
<td>IDS_APPLICATION</td>
</tr>
<tr>
<td>applicationid</td>
<td>A constructed attribute with values of the attributes MX_APPLICATION_ID and MX_REPOSITORYNAME (in cases where the MX_APPLICATION_ID does not exist). It may later be used in the listprivileges command and/or other operations where filtering on the application is possible.</td>
<td>AD</td>
</tr>
</tbody>
</table>
Example

SPML request (no filtering)
The following command lists all applications in the default system (observe the absence of @<systemid>):

```xml
<SOAP-ENV:Body>
  <spml:searchRequest xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"
    xmlns:spml="urn:oasis:names:tc:SPML:1:0">
    <spml:searchBase type="urn:oasis:names:tc:SPML:1:0#GUID">
      <spml:id>operation=listapplications</spml:id>
    </spml:searchBase>
    <dsml:filter>
      <dsml:present name="objectclass"></dsml:present>
    </dsml:filter>
  </spml:searchRequest>
</SOAP-ENV:Body>
```

SPML request (listing applications for one specific system)
The following command lists all applications in the system with systemid=idm1.

```xml
<SOAP-ENV:Body>
  <spml:searchRequest xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"
    xmlns:spml="urn:oasis:names:tc:SPML:1:0">
    <spml:searchBase type="urn:oasis:names:tc:SPML:1:0#GUID">
      <spml:id>operation=listapplications@idm1</spml:id>
    </spml:searchBase>
    <dsml:filter>
      <dsml:present name="objectclass"></dsml:present>
    </dsml:filter>
  </spml:searchRequest>
</SOAP-ENV:Body>
```

SPML response
The SPML response from the first operation (list applications from default system).

```xml
<SOAP-ENV:Body>
  <spml:searchResponse result="urn:oasis:names:tc:SPML:1:0#success"
    xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"
    xmlns:spml="urn:oasis:names:tc:SPML:1:0">
    <searchResultEntry>
      <spml:identifier>
        <spml:id>cn=ABAP2,operation=listapplications</spml:id>
      </spml:identifier>
      <spml:attributes>
        <dsml:attr name="objectclass">
          <dsml:value type="xsd:string">IDS_APPLICATION</dsml:value>
        </dsml:attr>
        <dsml:attr name="applicationid">
          <dsml:value type="xsd:string">ABAP2</dsml:value>
        </dsml:attr>
      </spml:attributes>
    </searchResultEntry>
    <searchResultEntry>
      <spml:identifier>
        <spml:id>cn=ABAP1,operation=listapplications</spml:id>
      </spml:identifier>
      <spml:attributes>
        <dsml:attr name="objectclass">
          <dsml:value type="xsd:string">IDS_APPLICATION</dsml:value>
        </dsml:attr>
        <dsml:attr name="applicationid">
          <dsml:value type="xsd:string">ABAP1</dsml:value>
        </dsml:attr>
      </spml:attributes>
    </searchResultEntry>
  </spml:searchResponse>
</SOAP-ENV:Body>
```
Listing available privileges (listprivileges)

Each of the applications in the Identity Center manages a certain set of privileges. The Identity Services consumer can find out about available privileges using following commands.

Operation properties

<table>
<thead>
<tr>
<th>Key</th>
<th>Value/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation</td>
<td>SPML Search operation</td>
</tr>
<tr>
<td>Starting point</td>
<td>operation= listprivileges</td>
</tr>
<tr>
<td>Search type</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Attributes requested</td>
<td>* or any attribute subset (see available attributes in section Returned data on page 21).</td>
</tr>
<tr>
<td>Filter</td>
<td>(objectclass=*) or any valid filtering based on the privilege's attributes (except mskey, see general limitation on page 13). See sections Filtering privileges per system and Listing privileges for one specific application.</td>
</tr>
</tbody>
</table>

The operation listprivileges is a special case of the general operation list (see section Listing multiple entries (list) on page 37) with filter objectclass=MX_PRIVILEGE.

The rules about selection of the specific system for the operation can be applied here.
Filtering privileges per system

Supplying an SPML identifier that consists only of:

listprivileges

will list all privileges managed by the default system.

To list all privileges that are managed by a specific system create identifier by specifying the relevant system ID in the following way:

listprivileges@<relevant system id>

Or:

list@<relevant system id>

In the latter case, you should also use filter "(objectclass=MX_PRIVILEGE)". See section Listing multiple entries (list) on page 37 for more information about list command.

Listing privileges for one specific application

Use applicationid in order to list all privileges that are managed by specific applications.

An exact match to a specific application or any type of the substring search is allowed:

(applicationid =<suffix of the valid application id>)
(applicationid =<middle part of the valid application id>*)
(applicationid =<prefix of the valid application id>*)

System filtering and application filtering are used together, so make sure that your request returns what you expect.

In order to list all privileges that are managed by an application that belongs to a specific system (which is NOT the default system), you should execute the following operation:

Identifier = listprivileges@<relevant system id>
Filter = (applicationid=<application id>)

If you do not supply @<system id>, the Identity Services will look in the default system and since it does not manage the specific application – the call will return 0 entries.

Returned data

A list of entries is returned. The identifiers of the returned entries are of the form

cn = <mskeyvalue>, <System Naming Context>

where the system naming context is the one that was selected by executed listprivileges command.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Value/Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>objectclass</td>
<td>Entry type of the object.</td>
<td>MX_PRIVILEGE</td>
</tr>
<tr>
<td>cn</td>
<td>The display name of the privilege.</td>
<td>MX_PRIV:GRC1</td>
</tr>
<tr>
<td>applicationid</td>
<td>The application ID of the application the privilege is defined for.</td>
<td>ABAP1</td>
</tr>
<tr>
<td>mskey</td>
<td>Identity Center internal attribute.</td>
<td>28</td>
</tr>
<tr>
<td>mskeyvalue</td>
<td>Identity Center internal attribute.</td>
<td>MX_PRIV:GRC1</td>
</tr>
<tr>
<td>Attribute</td>
<td>Description</td>
<td>Value/Example</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>changenumber</td>
<td>System internal attribute on an entry type, used for modification of the entry type, i.e. each modification of the entry type is identified by a specific changenumber.</td>
<td>161</td>
</tr>
<tr>
<td>memberperson</td>
<td>The distinguished name of the person (that has the privilege).</td>
<td>Lisa Andersson</td>
</tr>
<tr>
<td>parentrole</td>
<td>The role IDs of the roles that the privilege belongs to.</td>
<td>Sysadmin user</td>
</tr>
<tr>
<td>repository</td>
<td>The repository definition defined for the privilege (MX_REPOSITORYNAME).</td>
<td>GRC</td>
</tr>
</tbody>
</table>

**Example**

**SPML request (no filtering)**

The following command lists all privileges in the default system (observe the absence of @<systemid>).

```xml
<SOAP-ENV:Body>
  <spml:searchRequest xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"
                       xmlns:spml="urn:oasis:names:tc:SPML:1:0">
    <spml:searchBase type="urn:oasis:names:tc:SPML:1:0#GUID">
      <spml:id>operation=listprivileges</spml:id>
    </spml:searchBase>
    <dsml:filter>
      <dsml:present name="objectclass"></dsml:present>
    </dsml:filter>
  </spml:searchRequest>
</SOAP-ENV:Body>
```

**SPML request (filtering, per system)**

The following command lists all privileges in a specific system (observe the presence of @<systemid>):

```xml
<SOAP-ENV:Body>
  <spml:searchRequest xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"
                       xmlns:spml="urn:oasis:names:tc:SPML:1:0">
    <spml:searchBase type="urn:oasis:names:tc:SPML:1:0#GUID">
      <spml:id>operation=listprivileges@idm1</spml:id>
    </spml:searchBase>
    <dsml:filter>
      <dsml:present name="objectclass"></dsml:present>
    </dsml:filter>
  </spml:searchRequest>
</SOAP-ENV:Body>
```

**SPML request (filtering, per managed system and application)**

The following command lists all privileges in a specific system (here: idm1) and specific managed applications (here: all applications starting with "AB", i.e. applicationid equals "AB*"), and with the selection of attributes (cn, mskeyvalue, mskey, changenumber and applicationid):

```xml
<SOAP-ENV:Body>
  <spml:searchRequest xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"
                       xmlns:spml="urn:oasis:names:tc:SPML:1:0">
    <spml:searchBase type="urn:oasis:names:tc:SPML:1:0#GUID">
      <spml:id>operation=listprivileges@idm1</spml:id>
    </spml:searchBase>
    <dsml:filter>
      <dsml:substrings name="applicationid">
        <dsml:initial>AB</dsml:initial>
      </dsml:substrings>
    </dsml:filter>
    <dsml:attributes>
      <dsml:attribute name="cn" />
      <dsml:attribute name="mskeyvalue" />
    </dsml:attributes>
  </spml:searchRequest>
</SOAP-ENV:Body>
```
SPML response

This is the response for the last request (system idm1 and applications starting with "AB", which here are ABAP1 and ABAP2):

```xml
<SOAP-ENV:Body>
    <searchResultEntry>
      <spml:identifier>
        <spml:id>cn=MX_PRIV:GRC3,ou=nwidm1,o=ids</spml:id>
      </spml:identifier>
      <spml:attributes>
        <dsml:attr name="cn">
          <dsml:value type="xsd:string">MX_PRIV:GRC3</dsml:value>
        </dsml:attr>
        <dsml:attr name="mskeyvalue">
          <dsml:value type="xsd:string">MX_PRIV:GRC3</dsml:value>
        </dsml:attr>
        <dsml:attr name="mskey">
          <dsml:value type="xsd:string">30</dsml:value>
        </dsml:attr>
        <dsml:attr name="changenumber">
          <dsml:value type="xsd:string">163</dsml:value>
        </dsml:attr>
        <dsml:attr name="applicationid">
          <dsml:value type="xsd:string">ABAP2</dsml:value>
        </dsml:attr>
      </spml:attributes>
    </searchResultEntry>
    <searchResultEntry>
      <spml:identifier>
        <spml:id>cn=MX_PRIV:GRC1,ou=nwidm1,o=ids</spml:id>
      </spml:identifier>
      <spml:attributes>
        <dsml:attr name="cn">
          <dsml:value type="xsd:string">MX_PRIV:GRC1</dsml:value>
        </dsml:attr>
        <dsml:attr name="mskeyvalue">
          <dsml:value type="xsd:string">MX_PRIV:GRC1</dsml:value>
        </dsml:attr>
        <dsml:attr name="mskey">
          <dsml:value type="xsd:string">28</dsml:value>
        </dsml:attr>
        <dsml:attr name="changenumber">
          <dsml:value type="xsd:string">161</dsml:value>
        </dsml:attr>
        <dsml:attr name="applicationid">
          <dsml:value type="xsd:string">ABAP1</dsml:value>
        </dsml:attr>
      </spml:attributes>
    </searchResultEntry>
    <searchResultEntry>
      <spml:identifier>
        <spml:id>cn=MX_PRIV:GRC2,ou=nwidm1,o=ids</spml:id>
      </spml:identifier>
      <spml:attributes>
        <dsml:attr name="cn">
          <dsml:value type="xsd:string">MX_PRIV:GRC2</dsml:value>
        </dsml:attr>
        <dsml:attr name="mskeyvalue">
          <dsml:value type="xsd:string">MX_PRIV:GRC2</dsml:value>
        </dsml:attr>
        <dsml:attr name="mskey">
          <dsml:value type="xsd:string">29</dsml:value>
        </dsml:attr>
        <dsml:attr name="changenumber">
          <dsml:value type="xsd:string">162</dsml:value>
        </dsml:attr>
      </spml:attributes>
    </searchResultEntry>
  </spml:searchResponse>
</SOAP-ENV:Body>
```
Listing available roles (listroles)

The Identity Services consumer can retrieve the information about the available roles using the following commands.

Operation properties

<table>
<thead>
<tr>
<th>Key</th>
<th>Value/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation</td>
<td>SPML Search operation</td>
</tr>
<tr>
<td>Starting point</td>
<td>operation= listroles</td>
</tr>
<tr>
<td>Search type</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Attributes requested</td>
<td>* or any attribute subset (see attributes available for this operation in section Returned data on page 25)</td>
</tr>
<tr>
<td>Filter</td>
<td>(objectclass=*) or any valid filtering based on the role's attributes (except mskey, see general limitation on page 13). See section Filtering roles per system below.</td>
</tr>
</tbody>
</table>

The operation listroles is a special case of the general operation list (see section Listing multiple entries (list) on page 37) with filter objectclass=MX_ROLE.

The rules about selection of the specific system for the operation can be applied here.
Filtering roles per system

Supplying an SPML identifier that consists only of:

```
listroles
```

will list all roles managed by the default system.

To list all roles that are managed by a specific system create identifier by specifying the relevant system ID in the following way:

```
listroles@<relevant system id>
```

Or:

```
list@<relevant system id>
```

In the latter case, you should also use filter "(objectclass=MX_ROLE)". See section Listing multiple entries (list) on page 37 for information about list command.

Returned data

A list of entries is returned. The identifiers of the returned entries are of the form

```
cn = <mskeyvalue>, <System Naming Context>
```

where the system naming context is the one that was selected by executed listroles command.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Value/Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>objectclass</td>
<td>Entry type of the object.</td>
<td>MX_ROLE</td>
</tr>
<tr>
<td>cn</td>
<td>The display name of the role.</td>
<td>MX_ROLE:Manager</td>
</tr>
<tr>
<td>mskey</td>
<td>Identity Center internal attribute.</td>
<td>34</td>
</tr>
<tr>
<td>mskeyvalue</td>
<td>Identity Center internal attribute.</td>
<td>MX_ROLE:Manager</td>
</tr>
<tr>
<td>changenumber</td>
<td>System internal attribute on an entry type, used for modification of the entry type, i.e. each modification of the entry type is identified by a specific changenumber.</td>
<td>187</td>
</tr>
<tr>
<td>memberperson</td>
<td>The distinguished name of the person (that has the role).</td>
<td>Lisa Andersson</td>
</tr>
<tr>
<td>parentrole</td>
<td>The role IDs of the roles that this role belongs to.</td>
<td>MX_ROLE:Management</td>
</tr>
<tr>
<td>childrole</td>
<td>The role IDs of this role's child roles.</td>
<td>MX_ROLE:ITManager</td>
</tr>
<tr>
<td>childprivilege</td>
<td>IDs of this role's privileges.</td>
<td>MX_PRIV:WD:TAB_MANAGE,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MX_PRIV:GRC1</td>
</tr>
</tbody>
</table>
Example

SPML request (no filtering)
The following command lists all roles in the default system (observe the absence of @<systemid>).

```xml
<SOAP-ENV:Body>
  <spml:searchRequest xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"
    xmlns:spml="urn:oasis:names:tc:SPML:1:0">
    <spml:searchBase type="urn:oasis:names:tc:SPML:1:0#GUID">
      <spml:id>operation=listroles</spml:id>
    </spml:searchBase>
    <dsml:filter>
      <dsml:present name="objectclass"></dsml:present>
    </dsml:filter>
  </spml:searchRequest>
</SOAP-ENV:Body>
```

SPML request (filtering, per system)
The following command lists all roles in a specific system (observe the presence of @<systemid>, here: idm1) with the selected attributes (cn, mskeyvalue, mskey, changenumber and childprivilege):

```xml
<SOAP-ENV:Body>
  <spml:searchRequest xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"
    xmlns:spml="urn:oasis:names:tc:SPML:1:0">
    <spml:searchBase type="urn:oasis:names:tc:SPML:1:0#GUID">
      <spml:id>operation=listroles@idm1</spml:id>
    </spml:searchBase>
    <dsml:filter>
      <dsml:present name="objectclass"></dsml:present>
    </dsml:filter>
    <spml:attributes>
      <dsml:attribute name="cn" />
      <dsml:attribute name="mskeyvalue" />
      <dsml:attribute name="mskey" />
      <dsml:attribute name="changenumber" />
      <dsml:attribute name="childprivilege" />
    </spml:attributes>
  </spml:searchRequest>
</SOAP-ENV:Body>
```
This is the response for the last request (system idm1):

```xml
<soap-env:Body>
  <spml:searchResponse result="urn:oasis:names:tc:SPML:1:0#success">
    <searchResultEntry>
      <spml:identifier>
        <spml:id>cn=MX_ROLE:Manager,ou=nwidm1,o=ids</spml:id>
      </spml:identifier>
      <spml:attributes>
        <dsml:attr name="cn">
          <dsml:value type="xsd:string">MX_ROLE:Manager</dsml:value>
        </dsml:attr>
        <dsml:attr name="mskeyvalue">
          <dsml:value type="xsd:string">MX_ROLE:Manager</dsml:value>
        </dsml:attr>
        <dsml:attr name="mskey">
          <dsml:value type="xsd:string">34</dsml:value>
        </dsml:attr>
        <dsml:attr name="changenumber">
          <dsml:value type="xsd:string">187</dsml:value>
        </dsml:attr>
        <dsml:attr name="childprivilege">
          <dsml:value type="xsd:string">MX_PRIV:WD:TAB_HISTORY</dsml:value>
          <dsml:value type="xsd:string">MX_PRIV:WD:TAB_TODO</dsml:value>
          <dsml:value type="xsd:string">MX_PRIV:WD:TAB_MANAGE</dsml:value>
          <dsml:value type="xsd:string">MX_PRIV:WD:TAB_REPORT</dsml:value>
        </dsml:attr>
      </spml:attributes>
    </searchResultEntry>
    <searchResultEntry>
      <spml:identifier>
        <spml:id>cn=MX_ROLE:Developer,ou=nwidm1,o=ids</spml:id>
      </spml:identifier>
      <spml:attributes>
        <dsml:attr name="cn">
          <dsml:value type="xsd:string">MX_ROLE:Developer</dsml:value>
        </dsml:attr>
        <dsml:attr name="mskeyvalue">
          <dsml:value type="xsd:string">MX_ROLE:Developer</dsml:value>
        </dsml:attr>
        <dsml:attr name="mskey">
          <dsml:value type="xsd:string">33</dsml:value>
        </dsml:attr>
        <dsml:attr name="changenumber">
          <dsml:value type="xsd:string">181</dsml:value>
        </dsml:attr>
      </spml:attributes>
    </searchResultEntry>
  </spml:searchResponse>
</soap-env:Body>
```
Listing users (listusers)

The Identity Services consumer can retrieve information about users using following commands.

Operation properties

<table>
<thead>
<tr>
<th>Key</th>
<th>Value/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation</td>
<td>SPML Search operation</td>
</tr>
<tr>
<td>Starting point</td>
<td>operation= listusers</td>
</tr>
<tr>
<td>Search type</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Attributes requested</td>
<td>* or any attribute subset (see attributes available for this operation in section Returned data on page 29)</td>
</tr>
<tr>
<td>Filter</td>
<td>(objectclass=*) or any valid filtering based on the privilege's attributes (except mskey, see general limitation on page 13). See section Filtering users per system below.</td>
</tr>
</tbody>
</table>

The operation listusers is a special case of the general operation list (see section Listing multiple entries (list) on page 37) with filter objectclass=MX_PERSON.

The rules about selection of the specific system for the operation can be applied here.

Filtering users per system

Supplying an SPML identifier that consists only of:

listusers

will list all roles managed by the default system.

To list users that exist in a specific system create identifier by specifying the relevant system ID in the following way:

listusers@<relevant system id>

Or:

list@<relevant system id>

In the latter case, you should also use filter "(objectclass=MX_PERSON)". See section Listing multiple entries (list) on page 37 for information about list command.
Returned data

A list of entries is returned. The identifiers of the returned entries are of the form

\[\text{cn} = \text{<mskeyvalue>}, \text{<System Naming Context>}\]

where the system naming context is the one that was selected by executed listusers command.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Value/Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>objectclass</td>
<td>Entry type of the object.</td>
<td>MX_PERSON</td>
</tr>
<tr>
<td>cn</td>
<td>The display name of the privilege.</td>
<td></td>
</tr>
<tr>
<td>mskey</td>
<td>Identity Center internal attribute.</td>
<td></td>
</tr>
<tr>
<td>mskeyvalue</td>
<td>Identity Center internal attribute.</td>
<td></td>
</tr>
<tr>
<td>changenumber</td>
<td>System internal attribute on an entry type, used for modification of the entry type, i.e. each modification of the entry type is identified by a specific changenumber.</td>
<td></td>
</tr>
<tr>
<td>privilege</td>
<td>Privileges assigned to the user</td>
<td></td>
</tr>
<tr>
<td>role</td>
<td>Roles assigned to the user</td>
<td></td>
</tr>
</tbody>
</table>

In addition, user attributes are listed as the returned data. See section Returned data for operation schema (retrieving the schema) on page 32.

Example

SPML request (no filtering)

The following command lists all users in the default system (observe the absence of @<systemid>).

```xml
<SOAP-ENV:Body>
  <spml:searchRequest xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"
   xmlns:spml="urn:oasis:names:tc:SPML:1:0">
    <spml:searchBase type="urn:oasis:names:tc:SPML:1:0#GUID">
      <spml:id>operation=listusers</spml:id>
    </spml:searchBase>
    <dsml:filter>
      <dsml:present name="objectclass"></dsml:present>
    </dsml:filter>
  </spml:searchRequest>
</SOAP-ENV:Body>
```
SPML request (filtering, per system)
The following command lists all users in a specific system (observe the presence of @<systemid>, here: *idm1*) with the selected attributes (*mskeyvalue*, *cn*, *mskey*, *changenumber*, *role* and *privilege*):

```xml
<SOAP-ENV:Body>
  <spml:searchRequest xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"
    xmlns:spml="urn:oasis:names:tc:SPML:1:0">
    <spml:searchBase type="urn:oasis:names:tc:SPML:1:0#GUID">
      <spml:id>operation=listusers@idm1</spml:id>
    </spml:searchBase>
    <dsml:filter>
      <dsml:present name="objectclass"></dsml:present>
    </dsml:filter>
    <spml:attributes>
      <dsml:attribute name="mskeyvalue" />
      <dsml:attribute name="cn" />
      <dsml:attribute name="mskey" />
      <dsml:attribute name="changenumber" />
      <dsml:attribute name="role" />
      <dsml:attribute name="privilege" />
    </spml:attributes>
  </spml:searchRequest>
</SOAP-ENV:Body>
```

SPML request (filtering, per system, selected users)
The following command lists selected users (with *cn* starting with "L") in a specific system (here: *idm1*) with the selected attributes (*mskeyvalue*, *cn*, *mskey*, *changenumber*, *role* and *privilege*):

```xml
<SOAP-ENV:Body>
  <spml:searchRequest xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"
    xmlns:spml="urn:oasis:names:tc:SPML:1:0">
    <spml:searchBase type="urn:oasis:names:tc:SPML:1:0#GUID">
      <spml:id>operation=listusers@idm1</spml:id>
    </spml:searchBase>
    <dsml:filter>
      <dsml:substrings name="cn">
        <dsml:initial>L</dsml:initial>
      </dsml:substrings>
    </dsml:filter>
    <spml:attributes>
      <dsml:attribute name="mskeyvalue" />
      <dsml:attribute name="cn" />
      <dsml:attribute name="mskey" />
      <dsml:attribute name="changenumber" />
      <dsml:attribute name="role" />
      <dsml:attribute name="privilege" />
    </spml:attributes>
  </spml:searchRequest>
</SOAP-ENV:Body>
```
SPML response (filtering, per system, selected users)

This is the response for the last request, i.e. selected users (with `cn` starting with "L") in a specific system (here: `idm1`) with the selected attributes (`mskeyvalue`, `cn`, `mskey`, `changenumber`, `role` and `privilege`):

```xml
<SOAP-ENV:Body>
<searchResultEntry>
<spml:identifier>
<spml:id>cn=Lisa Andersson,ou=nwidm1,o=ids</spml:id>
</spml:identifier>
<spml:attributes>
<dsml:attr name="mskeyvalue">
<dsml:value type="xsd:string">Lisa Andersson</dsml:value>
</dsml:attr>
<dsml:attr name="cn">
<dsml:value type="xsd:string">Lisa Andersson</dsml:value>
</dsml:attr>
<dsml:attr name="mskey">
<dsml:value type="xsd:string">42</dsml:value>
</dsml:attr>
<dsml:attr name="changenumber">
<dsml:value type="xsd:string">205</dsml:value>
</dsml:attr>
<dsml:attr name="role" />
<dsml:attr name="privilege">
<dsml:value type="xsd:string">MX_PRIV:WD:TAB_TODO</dsml:value>
<dsml:value type="xsd:string">MX_PRIV:WD:TAB_MANAGE</dsml:value>
<dsml:value type="xsd:string">MX_PRIV:WD:TAB_HISTORY</dsml:value>
<dsml:value type="xsd:string">MX_PRIV:WD:TAB_REPORT</dsml:value>
<dsml:value type="xsd:string">MX_PRIV:CONFIG_AUDIT</dsml:value>
<dsml:value type="xsd:string">MX_PRIV:CONFIG_RW</dsml:value>
<dsml:value type="xsd:string">MX_PRIV:TRANSPORT:EXPORT</dsml:value>
<dsml:value type="xsd:string">MX_PRIV:TRANSPORT:IMPORT</dsml:value>
</dsml:attr>
</spml:attributes>
</searchResultEntry>
<searchResultEntry>
<spml:identifier>
<spml:id>cn=Lee Corser,ou=nwidm1,o=ids</spml:id>
</spml:identifier>
<spml:attributes>
<dsml:attr name="mskeyvalue">
<dsml:value type="xsd:string">Lee Corser</dsml:value>
</dsml:attr>
<dsml:attr name="cn">
<dsml:value type="xsd:string">Lee Corser</dsml:value>
</dsml:attr>
<dsml:attr name="mskey">
<dsml:value type="xsd:string">46</dsml:value>
</dsml:attr>
<dsml:attr name="changenumber">
<dsml:value type="xsd:string">227</dsml:value>
</dsml:attr>
<dsml:attr name="role">
<dsml:value type="xsd:string">MX_ROLE:Developer</dsml:value>
</dsml:attr>
<dsml:attr name="privilege">
<dsml:value type="xsd:string">MX_PRIV:GRC3</dsml:value>
</dsml:attr>
</spml:attributes>
</searchResultEntry>
</spml:searchResponse>
</SOAP-ENV:Body>
```
Retrieving the schema (schema)

It is possible to obtain schema by in two different ways:

- Executing an SPML getSchema request.
- Executing an SPML search request.

Operation properties

The SPML search request has the following properties:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation</td>
<td>SPML Search operation</td>
</tr>
<tr>
<td>Starting point</td>
<td>operation=schema</td>
</tr>
<tr>
<td>Search type</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Attributes requested</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Filter</td>
<td>Not applicable</td>
</tr>
</tbody>
</table>

Retrieving schema for a specific system

Supplying an SPML identifier that consists only of:

    operation=schema

will return the schema of the default system.

To obtain schema for a specific system create identifier by specifying the relevant system ID in the following way:

    operation=schema@<relevant system id>
    or
    operation=schema@<relevant system’s naming context>

Note:
The supplied system ID or naming context has to correspond to the system with SAPNWIDM type.

Returned data

The schema that is supported by Identity Services is returned. Currently, the following attributes are present in the schema that can be used for provisioning:

    objectclass
    preferredLanguage
    cn
    description
    islocked
    userPassword
    autoprivilege
    sn
    givenName
    mail
telephoneNumber
homephone
mobile
facsimileTelephoneNumber
pager
Example

SPML request

```xml
<SOAP-ENV:Body>
    <spml:searchBase type="urn:oasis:names:tc:SPML:1:0#GUID">
      <spml:id>operation=schema</spml:id>
    </spml:searchBase>
    <dsml:filter>
      <dsml:present name="objectclass"></dsml:present>
    </dsml:filter>
  </spml:searchRequest>
</SOAP-ENV:Body>
```

SPML response

```xml
<SOAP-ENV:Body>
    <searchResultEntry>
      <spml:identifier>
        <spml:id>ou=nwidm1,o=ids</spml:id>
      </spml:identifier>
      <spml:attributes>
        <dsml:value type="xsd:string">(repository-oid NAME 'repository' DESC 'Repository name' SYNTAX String)</dsml:value>
        <dsml:value type="xsd:string">(applicationid-oid NAME 'applicationid' DESC 'Application Identifier' SYNTAX String)</dsml:value>
        <dsml:value type="xsd:string">(objectclass-oid NAME 'objectclass' DESC 'Type of entry' SYNTAX String)</dsml:value>
        <dsml:value type="xsd:string">(mskeyvalue-oid NAME 'mskeyvalue' DESC 'Unique entry identifier' SYNTAX String)</dsml:value>
        <dsml:value type="xsd:string">(preferredlanguage-oid NAME 'preferredlanguage' DESC 'User language' SYNTAX String)</dsml:value>
        <dsml:value type="xsd:string">(cn-oid NAME 'cn' DESC 'User friendly name' SYNTAX String)</dsml:value>
        <dsml:value type="xsd:string">(description-oid NAME 'description' DESC 'Entry description' SYNTAX String)</dsml:value>
        <dsml:value type="xsd:string">(islocked-oid NAME 'islocked' DESC 'Account is disabled' SYNTAX String)</dsml:value>
        <dsml:value type="xsd:string">(userpassword-oid NAME 'userpassword' DESC 'Login password' SYNTAX String)</dsml:value>
    </spml:attributes>
  </searchResultEntry>
</spml:searchResponse>
</SOAP-ENV:Body>
```
<dsml:value type="xsd:string">(autoprivilege-oid NAME 'autoprivilege' DESC 'This attribute holds automatically assigned privileges. This is maintained by the system and should never be updated by external applications.' SYNTAX String)</dsml:value>
<dsml:value type="xsd:string">(sn-oid NAME 'sn' DESC 'User last name' SYNTAX String)</dsml:value>
<dsml:value type="xsd:string">(givenname-oid NAME 'givenname' DESC 'User first name' SYNTAX String)</dsml:value>
<dsml:value type="xsd:string">(mail-oid NAME 'mail' DESC 'Primary e-mail address' SYNTAX String)</dsml:value>
<dsml:value type="xsd:string">(telephonenumber-oid NAME 'telephonenumber' DESC 'Primary telephone number' SYNTAX String)</dsml:value>
<dsml:value type="xsd:string">(homephone-oid NAME 'homephone' DESC 'Additional telephone numbers' SYNTAX String)</dsml:value>
<dsml:value type="xsd:string">(mobile-oid NAME 'mobile' DESC 'Primary mobile number' SYNTAX String)</dsml:value>
<dsml:value type="xsd:string">(facsimiletelephonenumber-oid NAME 'facsimiletelephonenumber' DESC 'Primary fax number' SYNTAX String)</dsml:value>
<dsml:value type="xsd:string">(pager-oid NAME 'pager' DESC 'Additional pager numbers' SYNTAX String)</dsml:value>
<dsml:value type="xsd:string">(usercertificate-oid NAME 'usercertificate' DESC 'Certificate' SYNTAX String)</dsml:value>
<dsml:value type="xsd:string">(title-oid NAME 'title' DESC 'Academic Title' SYNTAX String)</dsml:value>
<dsml:value type="xsd:string">(initials-oid NAME 'initials' DESC 'Initials' SYNTAX String)</dsml:value>
<dsml:value type="xsd:string">(ou-oid NAME 'ou' DESC 'Department' SYNTAX String)</dsml:value>
<dsml:value type="xsd:string">(street-oid NAME 'street' DESC 'Address' SYNTAX String)</dsml:value>
<dsml:value type="xsd:string">(l-oid NAME 'l' DESC 'City' SYNTAX String)</dsml:value>
<dsml:value type="xsd:string">(postalcode-oid NAME 'postalcode' DESC 'Postal Code' SYNTAX String)</dsml:value>
<dsml:value type="xsd:string">(postofficebox-oid NAME 'postofficebox' DESC 'PO Box' SYNTAX String)</dsml:value>
<dsml:value type="xsd:string">(st-oid NAME 'st' DESC 'Region' SYNTAX String)</dsml:value>
<dsml:value type="xsd:string">(refrole-oid NAME 'refrole' DESC 'Reference to entry type MX_ROLE' SYNTAX String)</dsml:value>
<dsml:value type="xsd:string">(childrole-oid NAME 'childrole' DESC 'Member reference to entry type MX_ROLE' SYNTAX String)</dsml:value>
<dsml:value type="xsd:string">(autorole-oid NAME 'autorole' DESC 'This attribute holds automatically assigned roles. This is maintained by the system and should never be updated by external applications.' SYNTAX String)</dsml:value>
<dsml:value type="xsd:string">(privilege-oid NAME 'privilege' DESC 'Reference to entry type MX_PRIVILEGE' SYNTAX String)</dsml:value>
<dsml:value type="xsd:string">(childprivilege-oid NAME 'childprivilege' DESC 'Member reference to entry type MX_PRIVILEGE' SYNTAX String)</dsml:value>
<dsml:value type="xsd:string">(memberperson-oid NAME 'memberperson' DESC 'Member reference to entry type MX_PERSON' SYNTAX String)</dsml:value>
<dsml:value type="xsd:string">(group-oid NAME 'group' DESC 'Reference to entry type MX_GROUP' SYNTAX String)</dsml:value>
<dsml:value type="xsd:string">(membergroup-oid NAME 'membergroup' DESC 'Member reference to entry type MX_GROUP' SYNTAX String)</dsml:value>
Obtaining the result of a provisioning operation (requeststatus/auditlog)

The status of a provisioning operation is checked by executing the special Identity Services operation requeststatus. Operation auditlog, although still available, is obsolete and not recommended to use. See page 40 for information about the SPML request's requestID field that is used in this context.

Operation properties

The preferred operation requeststatus:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation</td>
<td>SPML Search operation</td>
</tr>
<tr>
<td>Starting point</td>
<td>operation=requeststatus</td>
</tr>
<tr>
<td>Search type</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Attributes requested</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Filter</td>
<td>(requestid=&lt;requestid&gt;)</td>
</tr>
</tbody>
</table>

And the obsolete operation auditlog:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation</td>
<td>SPML Search operation</td>
</tr>
<tr>
<td>Starting point</td>
<td>operation=auditlog</td>
</tr>
<tr>
<td>Search type</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Attributes requested</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Filter</td>
<td>(requestid=&lt;requestid&gt;)</td>
</tr>
</tbody>
</table>

Returned entry

A list of entries for the operation requeststatus is returned. The identifiers of the returned entries are on the form:

\[ cn = <mskeyvalue>, <System Naming Context> \]

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Value/Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>assigneditems</td>
<td>Pipe-separated list of roles (business roles) and/or privileges (technical roles) assigned to the entry.</td>
<td>ITEMID=MX_ROLE:Manager, STATUS=FAILED</td>
</tr>
<tr>
<td>status</td>
<td>The status of the request.</td>
<td>One of PENDING/APPROVED/REJECTED/FAILED/PARTIALLY APPROVED</td>
</tr>
<tr>
<td>requestid</td>
<td>The ID of the request.</td>
<td>1234</td>
</tr>
</tbody>
</table>
Example

SPML request

Supplied identifier:

\[ \text{Operation}=\text{requeststatus} \]

Supplied filter:

\[ \text{Requestid}=1234 \]

SOAP-ENV:Body>

    <spml:searchBase type="urn:oasis:names:tc:SPML:1:0#GUID">
        <spml:id>operation=\text{requeststatus}</spml:id>
    </spml:searchBase>
    <dsml:filter>
        <dsml:equalityMatch name="requestid">
            <dsml:value>1234</dsml:value>
        </dsml:equalityMatch>
    </dsml:filter>
</spml:searchRequest>
</SOAP-ENV:Body>

SPML response

<SOAP-ENV:Body>
        <searchResultEntry>
            <spml:identifier>
                <spml:id>cn=1234,operation=\text{requeststatus}</spml:id>
            </spml:identifier>
            <spml:attributes>
                <dsml:attr name="assignedroles">
                    ITEMID=MX\_ROLE:Manager,STATUS=FAILED|ITEMID=MX\_ROLE:Developer,STATUS=FAILED
                </dsml:attr>
                <dsml:attr name="status">
                    FAILED
                </dsml:attr>
                <dsml:attr name="requestid">
                    1234
                </dsml:attr>
            </spml:attributes>
        </searchResultEntry>
    </spml:searchResponse>
</SOAP-ENV:Body>
Obtaining the result of a role assignment
(roleassignmentresult)

The status of a role assignment is obtained by executing the special Identity Services operation roleassignmentresult.

Operation properties

The operation roleassignmentresult:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation</td>
<td>SPML Add operation</td>
</tr>
<tr>
<td>DN</td>
<td>operation=roleassignmentresult</td>
</tr>
<tr>
<td>Attributes</td>
<td>See the list of attributes in the section Attributes below.</td>
</tr>
</tbody>
</table>

Attributes

A list of attributes for the operation roleassignmentresult:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Value/Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>assigneditems</td>
<td>Comma- separated list of the items involved in the role assignment.</td>
<td>USER=TKB4053,CONNECTOR=GF7CLNT500, ITEM_NAME=TKB4053,ITEM_TYPE=USR, ITEM_ACTION=001,APPROVAL_STATUS= Approved,PROV_STATUS=4,COMMENTS= Output device PRNT does not exist</td>
</tr>
<tr>
<td>status</td>
<td>The status of the assignment.</td>
<td>One of SUCCESS/FAILURE/PARTIAL_SUCCESS</td>
</tr>
<tr>
<td>reason</td>
<td>Reason for the role assignment result, e.g. why the status is &quot;FAILURE&quot; (why the role assignment was declined).</td>
<td></td>
</tr>
<tr>
<td>requestid</td>
<td>The ID of the request.</td>
<td>1234</td>
</tr>
</tbody>
</table>
Listing multiple entries (list)

A default SPML search request returns information about a single entry (see page 44). The Identity Services implements a special search operation *list* that makes it possible to list multiple entries.

Using the filters (objectclass=<entry type>), the general operation list can be used for retrieving information about different entries. E.g. to list the privileges with the general operation=list, the filter needs to be (objectclass=MX_PRIVILEGE).

Operations *listprivileges, listusers* and *listroles* are special cases of the general *list* operation.

Operation properties

<table>
<thead>
<tr>
<th>Key</th>
<th>Value/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation</td>
<td>SPML Search operation</td>
</tr>
<tr>
<td>Starting point</td>
<td>operation=list</td>
</tr>
<tr>
<td>Search type</td>
<td>Not applicable</td>
</tr>
<tr>
<td>Attributes requested</td>
<td>* or any attribute subset.</td>
</tr>
<tr>
<td>Filter</td>
<td>The following must be present in the filter: (objectclass=&lt;entry type&gt;). In addition the filter can contain any valid filtering based on objects' attributes (except mskey, see general limitation on page 13).</td>
</tr>
</tbody>
</table>

*Note:*
Any entry type can be listed, but the limitation in attributes returned for each entry type exists (returns the set of data source attributes defined by the template).
Provisioning operations

Identity Services accepts the three default SPML provisioning operations:

- Add
- Modify
- Delete

The following description describes how a provisioning request is processed in asynchronous mode.

1. A provisioning request is sent to Identity Services.
   The SPML request contains the request ID field. Typically, the requestor will set the value for this field.
   Among other things, Identity Service extracts the request ID from the request. If the value is not given by requestor, the Identity Services will generate a new value. The SPML response's request ID field will be set to this value.

2. Identity Services accepts the request and returns the preliminary "OK" to the requestor.

3. If the request does not result in any other subsequent provisioning operations in the back-end system, then the returned "OK" is final and there is no need to check the status using the operation requeststatus (the user is provisioned to the identity store).

4. Otherwise, as in case when requesting privileges and roles, the Identity Services processes the request by starting back-end provisioning. Typically, this involves multiple requests to managed applications, approvals etc. In this case, the requestor should check the status of the operation using the operation requeststatus and the request ID (either self-defined or constructed by the Identity Services). See page 36 for details about operation for checking the status of provisioning operation.
   Possibly, retry operations due to error conditions.

Adding a person entry

This is done by performing an SPML Add operation.

Operation properties

<table>
<thead>
<tr>
<th>Key</th>
<th>Value/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation</td>
<td>SPML Add operation</td>
</tr>
<tr>
<td>DN</td>
<td>The unique ID of the entry to be added. It will be stored in mskeyvalue attribute in the identity store.</td>
</tr>
<tr>
<td>Attributes</td>
<td>Any set of the attributes available for the MX_PERSON. The default set of attributes that is exposed in the Identity Services is listed on page 32. Currently, it is possible to add only person objects.</td>
</tr>
</tbody>
</table>

All rules about identifier pre-processing, summarized on page 8 are applicable.
Example

SPML request

Supplied identifier:

Brad Jackson

Supplied attributes and values:

givenname=Brad, sn=Jackson, objectclass=MX_PERSON, mail=brad.jackson@example.com, privilege=MX_PRIV:GRC3, role=MX_ROLE:Manager

<SOAP-ENV:Body>
  <spml:addRequest requestID="1234"
xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"
xmlns:spml="urn:oasis:names:tc:SPML:1:0">
    <spml:identifier type="urn:oasis:names:tc:SPML:1:0#GUID">
      <spml:id>Brad Jackson</spml:id>
    </spml:identifier>
    <spml:attributes>
      <dsml:attr name="mail">
        <dsml:value>brad.jackson@example.com</dsml:value>
      </dsml:attr>
      <dsml:attr name="sn">
        <dsml:value>Jackson</dsml:value>
      </dsml:attr>
      <dsml:attr name="privilege">
        <dsml:value>MX_PRIV:GRC3</dsml:value>
      </dsml:attr>
      <dsml:attr name="givenname">
        <dsml:value>Brad</dsml:value>
      </dsml:attr>
      <dsml:attr name="role">
        <dsml:value>MX_ROLE:Manager</dsml:value>
      </dsml:attr>
      <dsml:attr name="objectclass">
        <dsml:value>MX_PERSON</dsml:value>
      </dsml:attr>
    </spml:attributes>
  </spml:addRequest>
</SOAP-ENV:Body>

SPML response (failure)

<SOAP-ENV:Body>
  <spml:addResponse result="urn:oasis:names:tc:SPML:1:0#failure"
xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"
xmlns:spml="urn:oasis:names:tc:SPML:1:0" requestID="1234">
  </spml:addResponse>
</SOAP-ENV:Body>

SPML response (success)

<SOAP-ENV:Body>
  <spml:addResponse result="urn:oasis:names:tc:SPML:1:0#success"
xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"
xmlns:spml="urn:oasis:names:tc:SPML:1:0" requestID="1234" />
</SOAP-ENV:Body>
Modifying a person entry

This is done by performing an SPML Modify operation.

Operation properties

<table>
<thead>
<tr>
<th>Key</th>
<th>Value/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation</td>
<td>SPML Modify operation.</td>
</tr>
<tr>
<td>DN</td>
<td>The unique ID of the entry to be modified.</td>
</tr>
<tr>
<td>Attributes</td>
<td>Any set of the attributes available for the MX_PERSON. The default set of attributes that is exposed in the identity Services is listed on page 32. Currently, it is possible to modify only person objects.</td>
</tr>
</tbody>
</table>

All rules about identifier pre-processing, summarized on page 8 are applicable.

Example

SPML request

Supplied identifier:

    Brad Jackson

Supplied attributes and values:

    mobile=+4790919095

    <SOAP-ENV:Body>
      <spml:modifyRequest xmlns:spml="urn:oasis:names:tc:SPML:1:0"
        xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"
        requestID="1235">
        <spml:identifier type="urn:oasis:names:tc:SPML:1:0#GUID">
          <spml:id>Brad Jackson</spml:id>
        </spml:identifier>
        <spml:modifications>
          <dsml:modification name="mobile" operation="add">
            <dsml:value>+4790919095</dsml:value>
          </dsml:modification>
        </spml:modifications>
      </spml:modifyRequest>
    </SOAP-ENV:Body>
Deleting a person entry

This is done by performing a SPML Delete operation.

Operation properties

<table>
<thead>
<tr>
<th>Key</th>
<th>Value/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation</td>
<td>SPML Delete operation</td>
</tr>
<tr>
<td>Starting point</td>
<td>The unique id of the entry to be deleted.</td>
</tr>
</tbody>
</table>

All rules about identifier pre-processing, summarized on page 8 are applicable.

Example

SPML request

Supplied identifier:

Brad Jackson

```xml
<SOAP-ENV:Body>
  <spml:deleteRequest requestID="1236"
    xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"
    xmlns:spml="urn:oasis:names:tc:SPML:1:0">
    <spml:identifier type="urn:oasis:names:tc:SPML:1:0#GUID">
      <spml:id>Brad Jackson</spml:id>
    </spml:identifier>
  </spml:deleteRequest>
</SOAP-ENV:Body>
```
Search operations

At any time it is possible to obtain information about entries managed by the Identity Services. The SPML search operation returns information about a single entry. To list multiple entries, use the special Identity Services operation list described on page 37.

Operation properties

<table>
<thead>
<tr>
<th>Key</th>
<th>Value/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation</td>
<td>SPML Search operation</td>
</tr>
<tr>
<td>Starting point</td>
<td>The unique ID of the entry to be listed.</td>
</tr>
<tr>
<td>Search type</td>
<td>Not applicable.</td>
</tr>
<tr>
<td>Attributes requested</td>
<td>* or any valid attribute subset .</td>
</tr>
<tr>
<td>Filter</td>
<td>(objectclass=*) or any valid filtering based on the object's attributes (except mskey, see general limitation on page 13).</td>
</tr>
</tbody>
</table>

Example

SPML request

```
<SOAP-ENV:Body>
  <spml:searchRequest xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"
                        xmlns:spml="urn:oasis:names:tc:SPML:1:0">
    <spml:searchBase type="urn:oasis:names:tc:SPML:1:0#GUID">
      <spml:id>Brad Jackson</spml:id>
    </spml:searchBase>
    <dsml:filter>
      <dsml:present name="objectclass"></dsml:present>
    </dsml:filter>
  </spml:searchRequest>
</SOAP-ENV:Body>
```
SPML response (entry after successful ADD)

This example shows SPML response after the first example Add operation (see page 40).

```xml
<SOAP-ENV:Body>
  <spml:searchResponse result="urn:oasis:names:tc:SPML:1:0#success"
 xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"
 xmlns:spml="urn:oasis:names:tc:SPML:1:0">
    <searchResultEntry>
      <spml:identifier>
        <spml:id>cn=Brad Jackson,ou=idm1,o=ids</spml:id>
      </spml:identifier>
      <spml:attributes>
        <dsml:attr name="sn">
          <dsml:value type="xsd:string">Jackson</dsml:value>
        </dsml:attr>
        <dsml:attr name="objectclass">
          <dsml:value type="xsd:string">MX_PERSON</dsml:value>
        </dsml:attr>
        <dsml:attr name="mskeyvalue">
          <dsml:value type="xsd:string">Brad Jackson</dsml:value>
        </dsml:attr>
        <dsml:attr name="mskey">
          <dsml:value type="xsd:string">50196</dsml:value>
        </dsml:attr>
        <dsml:attr name="givenname">
          <dsml:value type="xsd:string">Brad</dsml:value>
        </dsml:attr>
        <dsml:attr name="mail">
          <dsml:value type="xsd:string">brad.jackson@example.com</dsml:value>
        </dsml:attr>
        <dsml:attr name="privilege">
          <dsml:value type="xsd:string">MX_PRIV:GRC3</dsml:value>
        </dsml:attr>
        <dsml:attr name="role">
          <dsml:value type="xsd:string">MX_ROLE:Manager</dsml:value>
        </dsml:attr>
      </spml:attributes>
    </searchResultEntry>
  </spml:searchResponse>
</SOAP-ENV:Body>
```
SPML response (entry after successful Modify)

This example shows SPML response after the first example Modify operation (see page 42).

```xml
<SOAP-ENV:Body>
  <spml:searchResponse result="urn:oasis:names:tc:SPML:1:0#success">
    <spml:searchResultEntry>
      <spml:identifier>
        <spml:id>cn=Brad Jackson,ou=nwidm1,o=ids</spml:id>
      </spml:identifier>
      <spml:attributes>
        <dsml:attr name="sn">
          <dsml:value type="xsd:string">Jackson</dsml:value>
        </dsml:attr>
        <dsml:attr name="changenumber">
          <dsml:value type="xsd:string">587046</dsml:value>
        </dsml:attr>
        <dsml:attr name="objectclass">
          <dsml:value type="xsd:string">MX_PERSON</dsml:value>
        </dsml:attr>
        <dsml:attr name="cn">
          <dsml:value type="xsd:string">Brad Jackson</dsml:value>
        </dsml:attr>
        <dsml:attr name="mobile">
          <dsml:value type="xsd:string">+4790919095</dsml:value>
        </dsml:attr>
        <dsml:attr name="mskey">
          <dsml:value type="xsd:string">50196</dsml:value>
        </dsml:attr>
        <dsml:attr name="mskeyvalue">
          <dsml:value type="xsd:string">Brad Jackson</dsml:value>
        </dsml:attr>
        <dsml:attr name="givenname">
          <dsml:value type="xsd:string">Brad</dsml:value>
        </dsml:attr>
        <dsml:attr name="mail">
          <dsml:value type="xsd:string">brad.jackson@example.com</dsml:value>
        </dsml:attr>
        <dsml:attr name="privilege">
          <dsml:value type="xsd:string">MX_PRIV:GRC3</dsml:value>
        </dsml:attr>
        <dsml:attr name="role">
          <dsml:value type="xsd:string">MX_ROLE:Manager</dsml:value>
        </dsml:attr>
      </spml:attributes>
    </spml:searchResultEntry>
  </spml:searchResponse>
</SOAP-ENV:Body>
```