HP Select Identity

Software Version: 4.13

Web Services Guide

Document Release Date: May 2007
Software Release Date: May 2007

Legal Notices

Warranty

Hewlett-Packard makes no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

A copy of the specific warranty terms applicable to your Hewlett-Packard product can be obtained from your local Sales and Service Office.

Restricted Rights Legend

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause in DFARS 252.227-7013.

Hewlett-Packard Company
United States of America

Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19(c)(1,2).

Copyright Notices

This product includes software developed by the Apache Software Foundation (http://www.apache.org/).

Select Identity uses software from the Apache Jakarta Project including:

- Commons-beanutils.
- Commons-collections.
- Commons-logging.
- Commons-digester.
- Commons-httpclient.
- Element Construction Set (ecs).
- Jakarta-poi.
- Jakarta-regexp.
- Logging Services (log4j).

Additional third party software used by Select Identity includes:

- JasperReports developed by SourceForge.
- iText (for JasperReports) developed by SourceForge.
- BeanShell.
- Xalan from the Apache XML Project.
- Xerces from the Apache XML Project.
- Java API for XML Processing from the Apache XML Project.

- SOAP developed by the Apache Software Foundation.
- JavaMail from SUN Reference Implementation.
- Java Secure Socket Extension (JSSE) from SUN Reference Implementation.
- Java Cryptography Extension (JCE) from SUN Reference Implementation.
- JavaBeans Activation Framework (JAF) from SUN Reference Implementation.
- OpenSPML Toolkit from OpenSPML.org.
- JGraph developed by JGraph.
- Hibernate from Hibernate.org.
- BouncyCastle engine for keystore management, bouncycastle.org.

Trademark Notices

Documentation Updates

This manual's title page contains the following identifying information:

- Software version number, which indicates the software version
- Document release date, which changes each time the document is updated
- Software release date, which indicates the release date of this version of the software

To check for recent updates, or to verify that you are using the most recent edition of a document, go to:

http://ovweb.external.hp.com/lpe/doc_serv/

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Support

Please visit the HP OpenView support Web site at:

http://www.hp.com/managementsoftware/support

This Web site provides contact information and details about the products, services, and support that HP OpenView offers.

HP OpenView online software support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support site to:

- Search for knowledge documents of interest
- Submit enhancement requests online
- Download software patches
- Submit and track progress on support cases
- Manage a support contract
- Look up HP support contacts
- Review information about available services
- Enter discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and log in. Many also require a support contract.

To find more information about access levels, go to:

http://www.hp.com/managementsoftware/access_level

To register for an HP Passport ID, go to:

http://www.managementsoftware.hp.com/passport-registration.html

Contents

1 Introduction

Provisioning and maintaining user accounts and system access in HP Select Identity is typically done via “out of the box” GUI features. When a system requires a customized method of provisioning, Select Identity Web Services provides a flexible substitute using an XML framework, using the Service Provisioning Markup Language (SPML) developed by the Oasis Open organization’s Provisioning Services Technical Committee (PSTC) to facilitate data exchange among service provisioning systems.

Web Services can be used to create request mechanisms that correspond to those provided in the Select Identity GUI, but which work in nonstandard settings, or with systems that require specialized configuration.

External systems can send Simple Object Access Protocol (SOAP) messages to Select Identity for user provisioning. For each request, Web Services, working on a Request/Response Paradigm, sends a response.

This Guide discusses Select Identity Web Services within the overall context of SPML version 1.0. It provides details on how to develop properly-formed requests and includes numerous sample requests.

Audience

This guide is written for developers who are using Web Services to provision User Management functionality. It is essential that you are familiar with the OASIS SPML 1.0 specification and with XML in general.

Web Services is a Provisioning Service Point (PSP), as defined by the OASIS SPML 1.0 Specification. Therefore, only element and attribute extensions used in Web Services are thoroughly documented in this guide. The OASIS SPML 1.0 Specification provides complete documentation of the standard elements and attributes, in addition to the concepts, frames of reference, and conventions established for SPML provisioning.

Refer to the SPML Overview for:

- More information about how HP OpenView Select Identity has implemented Web Services according to the SPML 1.0 standard.
- The URL of the OASIS SPML 1.0 Specification.

Functional Example

The following example illustrates how a Web Services request functions:

The Human Resources department at “Company X” relies on an enterprise resource planning (ERP) application to manage employees. When a new employee is hired, the HR department adds the employee to the system. However, the new hire will need email and network

accounts and access to the systems on which he will fulfill his job responsibilities. Select Identity Web Services can be used as a mechanism enabling the ERP application to send a request to provision the user. Then, Select Identity can create the necessary accounts and access privileges on Company X's systems according to the services defined for the user.

SPML Overview

Select Identity supports Web Services according to the OASIS SPML 1.0 specification, which defines the concepts, operations, and XML schema for XML-based provisioning using a request and response paradigm.

The OASIS SPML 1.0 Specification, as the foundation for Web Services, is an important reference document if you are developing, configuring, or supporting Web Services on Select Identity. It is recommended that you download a copy at the following URL:

http://www.oasis-open.org/committees/download.php/3032/cs-pstc-spml-core-1.0.pdf

Context

Select Identity Web Services enforces all business processes on operations based on the context of each operation. Context is specified by the Requesting Authority (RA) that has initiated the request using the Select Identity API, and by the Provisioning Service Target (PST), which is the service on which the action will take place. When Select Identity receives a request (via an SPML-compliant message), the context of the request is determined as follows:

- Users or administrators can initiate requests such as user account creation, profile updates, password changes, addition of new service memberships, or stage completion in an approval process.
- Sensitive fields such as passwords can be encrypted using the security settings from the Select Identity configuration.

The following illustrates how identifying information for the person originating the request is passed to the PST:

```
<operationalAttributes xmlns="">
   <attr name='urn:oasis:names:tc:SPML:1:0#UserIDAndOrDomainName'>
      <value>sisa</value>
   </attr>
   <attr name='urn:trulogica:concero:2.0#password'>
      <value>abc123</value>
   </attr>
   <attr name="urn:trulogica:concero:2.0#keyFields">
      <value>UserName</value>
   </attr>
   <attr name="urn:trulogica:concero:2.0#serviceName">
      <value>CS1</value>
   </attr>
</operationalAttributes>
```

- The context of the PST is obtained from the following attribute in the request:

```
urn:trulogica:concero:2.0#serviceName
```

- SPML 1.0 mandates use of the `requestID` attribute in requests and responses, so that each response can be matched with the corresponding request in asynchronous requests. Select Identity generates a unique internal ID for each request. It is not necessary for an externally-generated request ID to be unique, although it is good practice.

The `Keyfields` attribute, shown in the code sample above, is used in reconciliation requests and is the primary account key for adding secondary user IDs. The `ServiceName` attribute is optional; if it is not present, the request is a modify profile request, for modifying user profile attributes, as opposed to user service attributes

Encoding

All SPML files that contain requests sent to Web Services must be properly-formed XML and must conform to the SOAP protocol.

Standard and Extended Requests

The SPML 1.0 standard allows operational elements to be used in their original form or an extended form. The Operational Summary provides details of the elements that are supported in Select Identity in original and extended form.

Extended requests are used to perform the following Select Identity operations:

- Terminate user
- Change password
- Reset password
- Enable user
- Disable user
- Enable user on service
- Disable user on service
- Transfer account
- Get user permissions

Individual and Batch Requests

Many requests submitted via Web Services are for individual delegated or self-service operations, such as profile modifications, user service membership addition or deletion, user resource account addition or deletion, or service account attribute modifications.

Web Services allows multiple requests to be processed as a group via batch request processing. Any user provisioning operation in Select Identity can be performed using this method. A single batch request consists of a series of requests that are all of the same type.

Batch processing uses HTTP as its transfer mechanism. In general, this type of processing is more suitable for smaller numbers of requests (less than 1000). For large numbers of requests, Select Identity's bulk capability is more suitable.

The following Web Services batch request types are supported in Select Identity version 4.10:

- Add
- Modify
- Delete

Batch Password operations are not supported in this release.

Primary and Secondary Account Clusters

Select Identity supports additional, "secondary," user identities associated with a single, "primary" user account. This grouping of a primary user ID and one or more secondary IDs is known as a user cluster. The first time a secondary account is created for a primary user ID, Select Identity creates a user account cluster.

Operations involving account clusters are not supported in Web Services in Select Identity version 4.11.

Sample Requests

The Select Identity product CD contains a `samples` directory that contains several sample Web Services requests. These can serve as a basis for developing and implementing requests on a Select Identity system.

2 Operational Summary

This section summarizes the operations supported in Web Services SPML, and provides structural information about typical requests. It also specifies which Select Identity operations are *not* supported in Web Services.

User Provisioning

Most user management operations available in Select Identity can be performed using Web Services. Both self-service and delegated requests are supported.

The following is a list of supported operations for Select Identity version 4.11:

- Add a user (`<addRequest>`)
- Add a user on a composite service (`<addRequest>`)
- Enable or disable a user account (`<extendedRequest>`)
- Enable or disable service membership (`<extendedRequest>`)
- Modify user attributes (except passwords and context) and entitlements (`<modifyRequest>`)
- Delete a user from a service (`<deleteRequest>`)
- Retrieve information about one or more users (`<searchRequest>`)
- Reset or change a user's password (`<extendedRequest>`)
- Terminate a user (`<extendedRequest>`)
- Batch requests of similar request types

When retrieving a user (`<searchRequest>`), either a list of users or various details, including resource accounts with entitlements, can be retrieved for individual users. See Chapter 4, Supported Elements and Attributes, for more details.

Operations that are Not Supported

The following operations are not supported in Web Services:

- Search operations for services, resources, or any object other than user accounts.
- Batch requests of mixed request types.
- Operations other than user account management (such as resource, service, context, or role creation and deletion).

- Terminating a primary user account *without* also terminating all secondary accounts (termination of the primary account automatically terminates all secondaries).

Anatomy of a Web Services Request

Web Services requests are structured as explained in this section, which uses a simple `<addRequest>` example (adding a user account) to illustrate the main sections of each request.

SOAP Envelope

Each request is contained within a SOAP envelope, typically sent over HTTP, as follows:

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

Request Type and Identifier

The request type and its identifier follow the SOAP envelope declaration:

```
<addRequest requestID='12345' execution='urn:oasis:names:tc:
SPML:1:0#asynchronous'>
```

The identifier value is returned in the `<requestResponse>` and is therefore important for tracking and auditing requests. Request IDs are normally numeric.

Operational Attributes

Within the body of a request, the operational attributes are provided first. These attributes specify identifying information about the user originating the request, including the user name, password, and the services to which the request is targeted (`Default`, `Global`, `USA`, and `Texas` in the example below).

For an `<addRequest>`, the specified `serviceName` values are the services on which the user is being added. For a `<modifyRequest>`, these values are the services to which the attributes being modified or deleted belong. For a `<deleteRequest>`, the `serviceName` values are the services from which the user is being deleted.

The following example illustrates the operational attributes section of a Web Services request:

```
<operationalAttributes>
  <attr name='urn:oasis:names:tc:SPML:1:0#UserIDAndOrDomainName'>
    <value>sisa</value>
  </attr>
  <attr name='urn:trulogica:concero:2.0#password'>
    <value>abc123</value>
  </attr>
  <attr name='urn:trulogica:concero:2.0#serviceName'>
    <value>Default</value>
    <value>Global</value>
    <value>USA</value>
    <value>Texas</value>
  </attr>
</operationalAttributes>
```

Delegated and Self Service Requests

Self-service requests are submitted by a user on his or her own behalf. Delegated requests are performed for a user's account by a system administrator.

In the case of a delegated request, the user account from which the request is made must have an appropriate level of administration privileges to allow the request. Self-service requests are identical to delegated requests in every regard except that they are made by the user who owns the account, who often has a low level of access and administration privileges relative to accounts making delegated requests.

Request Attributes

The data to be used in the request follow the operational attributes as a series of individual `<attributes>`, such as the user's first and last names, email account, and password. Each attribute is bounded by the `<attr>` and `</attr>` tag.

All attributes, including user name, password, and entitlements, can be autogenerated by Select Identity.

The following example shows identity and profile information for a user named `AAllen`, together with entitlements for a resource named `LDAP`. Entitlements can also be presented using the format `ResourceName_Entitlements`:

```
<attributes>
   <attr name='UserName'>
      <value>AAllen</value>
   </attr>
   <attr name='Password'>
      <value>abcd1234</value>
   </attr>
   <attr name='FirstName'>
      <value>Anna</value>
   </attr>
   <attr name='LastName'>
      <value>Allen</value>
   </attr>
   <attr name='Email'>
      <value>anna.allen@companyx.com</value>
   </attr>
   <attr name='urn:trulogica:concero:2.0#groups:LDAP'>
      <value>Group 1</value>
      <value>Group 2</value>
   </attr>
   <attr name='Company'>
      <value>Company X</value>
   </attr>
   <attr name="Department">
      <value>Sales</value>
   </attr>
   <attr name='City'>
      <value>Dallas</value>
   </attr>
   <attr name='State'>
      <value>Texas</value>
   </attr>
   <attr name='Country'>
      <value>USA</value>
   </attr>
   <attr name='BirthDate'>
      <value>08/09/1980</value>
   </attr>
```

```
        <attr name="Zip"/>
    </attributes>
```

Request Closure

At the end of the attributes, the request is closed, as are the request body and SOAP envelope, as follows:

```
        </addRequest>
    </soap:Body>
</soap:Envelope>
```

Responses

SPML uses a *request-response paradigm.* For every type of `<Request>` there exists a correspondingly-named `<Response>`. Responses provide confirmation if the originating request was successful. They also provide information about the cause of failure; this can be interpreted to correct the source of the problem. In addition, a `<searchRequest>` returns records that match the search criteria by enclosing them in a `<searchResponse>`.

Reconciliation

You can set up Web Services to allow a user's attributes to be changed on a resource and then synchronized to propagate the change into the Select Identity repository.

This is called **reconciliation.** It enables a resource to push user information to Select Identity. The agent on the resource (provided by a two-way connector) tracks changes made on the resource, and synchronizes them with the data in Select Identity.

All attributes changed in Select Identity are pushed out to resources according to the attribute mapping that has been defined. For detailed information about mapping attributes, see the Select Identity online help for administration, including the section about the Attribute Mapper utility.

For additional information about reconciliation and authoritative versus non-authoritative resources, refer to the appropriate sections in the Select Identity Online Help for Administration and the *HP OpenView Select Identity Administration Guide.*

Reconciliation Rules

Rules identify the services that are affected when a reconciliation operation is issued, based on the resource ID sent by the resource. Rules also specify the location of the reverse mapping that should to be performed for add and modify operations, such as:

```
NT Domain Resource -> ntuser.properties.
```

Thus, reverse synchronization can be performed only for services with specified mapping for the incoming resource ID. If the mapping does not exist, the request is logged and ignored.

Reconciliation Operations in Web Services

The following reconciliation operations can be performed using Web Services, either individually or as a batch:

- Add a user, if the resource is authoritative (`<addRequest>`)
- Add attributes to an existing user, if the resource is non-authoritative (`<addRequest>` or `<modifyRequest>`)
- Modify user attributes and entitlements (`<modifyRequest>`)
- Delete a user (`<deleteRequest>`)
- Issue multiple add, modify, and/or delete requests (`<batchRequest>`)

Unlike batch requests, reconciliation requests can contain mixed request types; they are not limited to a single type of request.

Security

Select Identity provides the same level of security for Web Services as for the browser GUI. Thus, Select Identity supports HTTP through SSL 3.0 or TLS 1.0.

To enable external authentication of requests, it is necessary to modify properties in the `truAccess.properties` file. Refer to the *HP OpenView Select Identity Installation Guide* for details. This guide contains an appendix that lists all the properties in this file, in addition to a section that describes the Select Identity security framework and keystores.

Authentication and Authorization

Web Services uses the Select Identity authentication mechanism. To authenticate requests, Web Services supports basic user authentication via the Select Identity user name and password that must be included in the request, in an `<operationalAttributes>` element.

Before any operation is completed, the user name and password is verified against an authoritative resource. For authorization, Web Services allows or denies the request according the roles assigned to the user. The user can only perform operations according to the permissions set within the assigned roles.

External authentication can be achieved through use of a verification external call assigned to the `<Password>` attribute. Refer to the Select Identity online help for administration, which includes information about registering and using external calls. The Select Identity *External Call Developer Guide* provides more detailed and comprehensive information for developers to use in coding custom external calls.

Resource Identification For Reconciliation Requests

In the case of a modification request, you can add an optional resource identifier to indicate that the origin of the request is from an authoritative resource. This is primarily used by agent software to indicate changes in attribute values in a resource.

If the resource identifier is specified, an additional check is made to verify that the security principal is associated with the resource identified by the Resource ID. Aside from this, there is no difference in the SPML for a request from an authoritative versus a non-authoritative resource.

If Standard Web Services requests include resource identifiers, this data is ignored.

An initial user creation request must come from an authoritative resource. Once the user is created in Select Identity, resource specific entitlements can be added from non-authoritative resources.

For further information, refer to the Reconciliation chapter in the *Select Identity Online Help for Administration*.

Web Services and Multiple Resource IDs

This section describes Web Services user management operations in the context of the Select Identity multiple resource ID feature.

In Select Identity, a user is normally identified by a single user name. However, in some cases, it is desirable for a user to have several user names. For instance, one might be the person's general-purpose user name, and another might be a task and/or resource-specific user name. Multiple user IDs are organized so that one is the "primary" user name and all others are "secondary" user names. The person who owns these user names and accounts logs into Select Identity with the primary ID.

Delegated User Management and Multiple Resource IDs

The following operations are supported for multiple resource IDs in self-service and delegated modes:

- Add secondary user to service and associate with a primary user.

  When a secondary user is created, the `<operationalAttributes>` must include the primary user key for validation. This is not needed for other operations, as it is available to Select Identity when it is retrieved from the database.
- Modify secondary user in a service.
- Delete secondary user from service.
- Enable secondary user in a service.
- Disable secondary user in a service.
- Enable secondary user.
- Disable secondary user.
- Terminate secondary user.
- Transfer Accounts from one primary to another.
- View primary user service membership, and return a `<searchResponse>` that contains all secondary users.
- View primary user profile, and return a `<searchResponse>` that contains all secondary users.

- View primary user profile, and return a <searchResponse> that contains all secondary users, all service accounts, and all resource accounts.
- Reset or change user password.

Reconciliation and Multiple Resource IDs

Web Services provides the capability to:

- Link multiple resource accounts to one user account in a cluster consisting of one primary user and one or more secondary accounts, during reconciliation add or modify requests.
- Synchronize Select Identity user attributes across multiple accounts within a cluster based on the user's auto synchronization control flag settings in the reconciliation data file.

Password Management and Multiple Resource IDs

Use Web Services to change the password for supplied user accounts, in a similar fashion to self-service password changes. The user account for login *must* be the primary account, and only this account password is verified. The password change can be either for the primary account or for a specified secondary account.

The changePassword operation is designed for self service (and secondary users by the Primary). The resetPassword operation is only used in delegated password reset operations.

Password change requests are performed using the <extendedRequest> SPML element. They must include the <operationIdentifier> element with the <operationID> defined as changePassword, as in the following example:

```
<operationIdentifier operationIDType='urn:oasis:names:tc:SPML:1:0#URN'>
   <operationID>urn:trulogica:concero:2.0#changePassword</operationID>
</operationIdentifier>
```

For password modification requests, the <operationIdentifier> is as follows:

```
<operationIdentifier operationIDType='urn:oasis:names:tc:SPML:1:0#URN'>
   <operationID>urn:trulogica:concero:2.0#resetPassword</operationID>
</operationIdentifier>
```

3 Issuing Requests

To issue Web Services user provisioning and data synchronization requests to Select Identity, an external system must send a SOAP message containing a Service Provisioning Markup Language (SPML) request.

URL for Sending Requests

All Web Services requests must be sent to the following URL:

http://select_identity_host:port/lmz/webservice/

The section of the URL in italic type corresponds to the usual address or hostname where you access Select Identity.

Example Code

The code provided in this section is a simple example of a Java program that issues a call to Web Services hosted on the local host. The SPML payload is pulled from an external file. It also processes the response generated by the service after it receives the request.

```
import java.io.File;
import java.io.FileInputStream;

import org.apache.commons.httpclient.HttpClient;
import org.apache.commons.httpclient.methods.EntityEnclosingMethod;
import org.apache.commons.httpclient.methods.PostMethod;

public class Test
{
  // This is where you send all your SOAP(SPML) requests
  public static String MAIN_SERVICE_URL =
"http://192.168.1.20:7001/lmz/webservice/";

  public static void main(String args[])
  {
    try
    {
      Test t = new Test();

  System.out.println("Sending "+ args[0] + "to the server " +
args[1]);
  if (args[1] != null)
 MAIN_SERVICE_URL = args[1];
      t.sendXmlFile(args[0], MAIN_SERVICE_URL);   // Send XML messages
to server
    }
    catch(Exception excp) { excp.printStackTrace(); }
  }
```

```
  public void sendXmlFile(String fileName, String url) throws Exception
  {
    // Get file to be posted
    File input = new File(fileName);

System.out.println("Sending request to " + url);

    // Prepare HTTP post
    PostMethod post = new PostMethod(url);

    // Request content will be retrieved directly from the input stream
    post.setRequestBody(new FileInputStream(input));

    // Per default, the request content needs to be buffered in order to
determine its length.
    // Request body buffering can be avoided when
    // = content length is explicitly specified
    // = chunk-encoding is used
    if (input.length() < Integer.MAX_VALUE) {
        post.setRequestContentLength((int)input.length());
    } else {

post.setRequestContentLength(EntityEnclosingMethod.CONTENT_LENGTH_CHUNKED);
    }

    // Specify content type and encoding
    post.setRequestHeader("Content-type", "text/xml;
charset=ISO-8859-1");

    HttpClient httpclient = new HttpClient(); // Get HTTP client

    int result = httpclient.executeMethod(post); // Execute request

    System.out.println("Response status code: " + result);
    System.out.println("Response body: ");
    System.out.println(post.getResponseBodyAsString());

    post.releaseConnection(); // Release current connection to the
connection pool
  }
}
```

4 Supported Elements and Attributes

This section provides detailed discussion of the supported request elements and their attributes, illustrated with sample sections of SPML.

The HP OpenView Select Identity product CD contains a `samples` directory, where you will find a set of Web Services SPML request samples. These are provided so that you can use them as a basis for developing custom requests.

<addRequest>

Select Identity supports several variations of the `<addRequest>` element. In any `<addRequest>`, if the user name already exists in Select Identity, then other specified attributes, such as service memberships, will be updated. An `<addRequest>` only adds a user account if the specified account does not already exist in Select Identity. Since a `<modifyRequest>` only allows modification of profile attributes, the `<addRequest>` is used whenever the objective is to add a user account to a given service.

The following is a partial example of an `<addRequest>`. The `<operationalAttributes>` provide login credentials for the administrator or self-service user originating the request. If you add one or more services to the `serviceName` operational attribute (using a value that corresponds to one of the services in your Select Identity system), the `<addRequest>` subscribes the user to that service as well. The `<attributes>` element provides information about the user. These attributes are ServiceView-specific, because a given view may not have all service attributes; an attribute cannot be accessed if it is not in the ServiceView for the request type.

Elements and Attributes

```
<addRequest ... requestID=12345>
```

The ID of the provisioning transaction. This ID is present in the `<requestResponse>`.

```
<operationalAttributes>
   <attr name='urn:oasis:names:tc:SPML:1:0#UserIDAndOrDomainName'>
      <value>sysAdmin</value>
   </attr>
   <attr name='urn:trulogica:concero:2.0#password'>
      <value>pw1234</value>
   </attr>
   <attr name='urn:trulogica:concero:2.0#serviceName'>
      <value>08</value>
   </attr>
</operationalAttributes>
```

The username and password of the administrator authorized to add a user to the specified service, or the user name and password of a self-service user. Select Identity authenticates this user against the `serviceName` attribute.

You can provision a user onto more than one service by placing multiple values in the `serviceName` operational attribute.

Use these attributes in simple add requests or in add requests issued during reconciliation by an authoritative or non-authoritative resource. It can be omitted from self-service requests.

```
<operationalAttributes>
   <attr name='urn:trulogica:concero:2.0#reverseSync'>
      <value>true</value>
   </attr>
```

The `reverseSync` attribute indicates that the operation is a reverse synchronization (data is pushed from the resource to Select Identity). This value must always be true. Use this attribute for reverse synchronization or in add requests issued during reconciliation by an authoritative or non-authoritative resource.

```
<operationalAttributes>
   <attr name='urn:trulogica:concero:2.0#resourceId'>
      <value>resource_ID</value>
   </attr>
```

The ID of the resource issuing the request, as in the case of user import or synchronization of resource data. Use this attribute for reverse synchronization or in add requests issued during reconciliation by an authoritative or non-authoritative resource.

```
<operationalAttributes>
   <attr name='urn:trulogica:concero:2.0#keyFields'>
      <value>field</value>
   </attr>
```

The key field to update in the Select Identity database during reconciliation. Use the `keyfields` attribute in add requests issued during reconciliation by an authoritative or non-authoritative resource.

```
<operationalAttributes>
   <attr name='urn:trulogica:concero:2.0#taUserName'>
      <value>ID</value>
   </attr>
```

The name or ID of the user in Select Identity. Use this attribute for all reconciliation requests (standalone `<addRequest>` and `<addRequest>` in a `<batchRequest>`).

```
<operationalAttributes>
   <attr name='urn:trulogica:concero:2.0#taResourceKey'>
      <value>key</value>
   </attr>
```

The name or ID of the user on the resource. Specify this attribute for all reconciliation requests (standalone `<addRequest>` and `<addRequest>` in a `<batchRequest>`).

```
<attributes>
   <attr name='UserName'>
      <value>wsuser1</value>
   </attr>
   <attr name='Password'>
      <value>abcd1234</value>
   </attr>
   <attr name='FirstName'>
      <value>Anna</value>
   </attr>
   <attr name='LastName'>
      <value>Allen</value>
   </attr>
   <attr name='Email'>
      <value>user@companyx.com</value>
   </attr>
   <attr name='urn:trulogica:concero:2.0#groups:LDAP177'>
```

```
      <value>West Oregon</value>
      <value>West Washington</value>
    </attr>
    <attr name='Company'>
      <value>HP</value>
    </attr>
  </attributes>
```

The name-value pairs of the attributes to add for the user, where `attr_name` is the name of the attribute and `value` is the value. You must provide an `<attr>` element for each attribute required by the resource(s) for a newly created user. Do not specify attributes that are autogenerated.

To add a user with a system-generated username and password, use the following values for the `FirstName` and `LastName` attributes:

```
<attr name='FirstName'>
  <value>Username and Password</value>
</attr>
<attr name='LastName'>
  <value>Generation</value>
</attr>
```

Each attribute name is defined when an administrator creates a service in Select Identity. If performing reverse synchronization, the attribute names are defined by the resource, and Select Identity determines this based on rules.

If you wish to provide entitlements, include the attribute name `urn:trulogica:concero:2.0#groups:resource`, and provide a value element for each entitlement you wish to grant. If performing reverse synchronization, you do not need to specify the resource.

A service can be supported by multiple resources. The services a to which a resource is assigned filter out un-needed attributes.

When provisioning administrative user accounts, you can include attributes that enable the user to be provisioned with a specific administrative role and context.

The `<SIAdminRole>`, `<SIService>`, and `<RoleServiceContext>` attributes contain the assigned combination of the role name, the service on which the role exists, and the context.

These attributes can be used to assign any admin role to a user ID. The sample below would assign the `Concero Sys Admin` role on services named `Global` and `Default`. (This combination would only be provisioned onto an administrative user, since it confers a level of permission inappropriate to a non-administrative account.)

The `RoleServiceContext` is set to all contexts (using an asterisk as a wildcard for the context), for the assigned role on these services.

```
<attr name='SIAdminRole'>
  <value>Concero Sys Admin</value>
</attr>
<attr name='SIService'>
  <value>Global</value>
  <value>Default</value>
</attr>
<attr name='RoleServiceContext'>
  <value>Concero Sys Admin|Global|*</value>
  <value>Concero Sys Admin|Default|*</value>
</attr>
```

The following list demonstrates how the <SIAdminRole>, <SIService>, and <RoleServiceContext> attributes can be assign to a user account any combination of roles, services, and contexts. For example:

- **Specified service, specified context:** Admin Role 1|Service1|Company|HP
- **All services, any Company context:** Admin Role 1|*|Company|*
- **Specified service, all contexts:** Admin Role 1|Service1|*
- **All services, specified context:** Admin Role 1|*|Company|HP
- **All services, all contexts:** Admin Role 1|*|*

<deleteRequest>

Delete operations using the <deleteRequest> remove service subscriptions from user accounts that are otherwise left intact. They do not necessarily delete attributes or attribute values, although they do normally delete entitlement values for a given resource that a service provided.

User account removal must be performed using an <extendedRequest>, which terminates the account. Requests to disable user accounts service memberships are performed using the <extendedRequest>.

The following is a partial example of a <deleteRequest>. The <operationalAttributes> provide login credentials for the administrator or self-service user originating the request, and the service from which the user is to be deleted. The <attributes> element specifies information about the user. These attributes are service-view-specific.

Elements and Attributes

```
<deleteRequest ... requestID=12345>
```

The ID of the provisioning transaction. This ID is present in the <requestResponse>.

```
<operationalAttributes>
   <attr name='urn:oasis:names:tc:SPML:1:0#UserIDAndOrDomainName'>
      <value>admin_username</value>
   </attr>
   <attr name='urn:trulogica:concero:2.0#password'>
      <value>123passwd</value>
   </attr>
   <attr name='urn:trulogica:concero:2.0#serviceName'>
      <value>service_name</value>
   </attr>
```

The user name and password of the administrator authorized to delete the user, or that of a self-service user, followed by the name of the service on which the request is to be carried out.

```
<operationalAttributes>
<attr name='urn:trulogica:concero:2.0#reverseSync'>
<value>true</value>
</attr>
```

The reverseSync attribute indicates that the operation is a reverse synchronization (data is pushed from the resource to Select Identity). This value must always be true. Specify this attribute for reverse synchronization or in delete requests issued during reconciliation by an authoritative or non-authoritative resource.

```
<operationalAttributes>
<attr name='urn:trulogica:concero:2.0#keyFields'>
<value>field</value>
</attr>
```

The key field to update in the Select Identity database during reconciliation. Multiple `keyFields` can be included. Specify this in delete requests issued during reconciliation by an authoritative or non-authoritative resource.

```
<identifier identifierType='urn:oasis:names:tc:SPML:
1:0#UserIDAndOrDomainName'>
   <id>id</id>
</identifier>
```

The Select Identity ID of the user to delete.

<batchRequest>

The operational attributes of each request in a batch, by default, are taken from those specified at the beginning of the request. Alternatively, operational attributes can be specified for some or all of the requests in the batch.

To add accounts using a `<batchRequest>`, regardless of whether they are primary, secondary, or normal, you must have Add User permission on the service to which the request is targeted. A `<batchRequest>` from a user who only has Add Account permission fails.

The following is a partial example of a `<batchRequest>`.

```
<batchRequest ... requestID=value>
```

The ID of the provisioning transaction. This ID is present in the `<requestResponse>`.

```
<operationalAttributes>
   <attr name='urn:oasis:names:tc:SPML:1:0#UserIDAndOrDomainName'>
      <value>admin_username</value>
</attr>
<attr name='urn:trulogica:concero:2.0#password'>
   <value>admin_passwd</value>
</attr>
```

The user name and password of the administrator authorized to perform the requests. Select Identity authenticates this administrative user against the service.

```
<operationalAttributes>
   <attr name='urn:trulogica:concero:2.0#resourceId'>
      <value>resource_ID</value>
   </attr>
```

The ID of the resource issuing the request, as in the case of auto-discovery or synchronization of resource data.

```
<operationalAttributes>
   <attr name='urn:trulogica:concero:2.0#keyFields'>
      <value>field</value>
   </attr>
```

The attribute to use to idenitify the user in Select Identity and the resource.

```
<operationalAttributes>
   <attr name='urn:trulogica:concero:2.0#reverseSync'>
      <value>true</value>
   </attr>
```

Indicates that the operation is a reverse synchronization (data is pushed from the resource to Select Identity). This value must always be true.

```
<addRequest requestID=value>
   <operationalAttributes />
      <attributes>
         <attr name="username">
            <value>user1</value>
         </attr>
         ...
      </attributes>
</addRequest>
```

The format for each request to process. There may be one or more `<addRequest>`, `<modifyRequest>`, or `<deleteRequest>` elements in any order, and all of the same type.

<extendedRequest>

The `<extendedRequest>` element allows operations that are not explicitly part of the SPML 1.0 specification. Select Identity Web Services supports several operations using the `<extendedRequest>`.

The following samples illustrate operations performed using the `<extendedRequest>` element. Some examples of typical requests follow the summary of attributes and values.

Attributes and Values

```
<extendedRequest ... requestID=value>
```

The ID of the provisioning transaction. This ID is present in the `<requestResponse>`.

```
<operationalAttributes>
   <attr name='urn:oasis:names:tc:SPML:1:0#UserIDAndOrDomainName'>
      <value>admin_username</value>
   </attr>
```

The user name and password of the administrator authorized to add a user to the service, or that of a self-service user. Select Identity authenticates this user against the service.

```
<operationalAttributes>
   <attr name='urn:trulogica:concero:2.0#password'>
      <value>admin_passwd</value>
   </attr>
   <attr name='urn:trulogica:concero:2.0#serviceName'>
      <value>service_name</value>
   </attr>
```

The service where the user will be enabled or disabled in an `enableMembership` or `disableMembership` operation.

```
<operationalAttributes>
   <attr name='urn:trulogica:concero:2.0#reverseSync'>
      <value>true</value>
   </attr>
```

Indicates that the operation is a reverse synchronization (data is pushed from the resource to Select Identity). This value must always be true. Include this attribute for reverse synchronization only.

```
<operationalAttributes>
   <attr name='urn:trulogica:concero:2.0#resourceId'>
      <value>resource_ID</value>
   </attr>
```

The resource issuing the request, as in the case of auto-discovery or synchronization of resource data. Include this attribute for reverse synchronization only.

```
<identifier identifierType='urn:oasis:names:tc:SPML:
1:0#UserIDAndOrDomainName'>
   <id>ResourceID</id>
</identifier>
```

The user name to modify (may be the same as that in the `<operationalAttributes>` if this is a self-service request).

```
<providerIdentifier providerIDType='urn:oasis:names:tc:SPML:1:0#URN'>
   <providerID>urn:trulogica:concero:2.0</providerID>
</providerIdentifier>
```

The provider of the service (Select Identity).

```
<operationIdentifier operationIdType='urn:oasis:names:tc:SPML:1:0#URN'>
   <operationID>urn:trulogica:concero:2.0#operation</operationID>
</operationIdentifier>
```

The operation to perform. The following values are valid for the `<operationIdentifier>`, depending on the request:

- `enableMembership`: Enable a user's membership on the specified service:

```
<operationIdentifier operationIdType='urn:oasis:names:tc:SPML:1:0#URN'>
   <operationID>urn:trulogica:concero:2.0#enableMembership</operationID>
</operationIdentifier>
```

- `disableMembership`: disable a user's membership on the specified service:

```
<operationIdentifier operationIdType='urn:oasis:names:tc:SPML:1:0#URN'>
   <operationID>urn:trulogica:concero:2.0#disableMembership</operationID>
</operationIdentifier>
```

- `changePassword`: change a user's password (for self-service use only):

```
<operationIdentifier operationIdType='urn:oasis:names:tc:SPML:1:0#URN'>
   <operationID>urn:trulogica:concero:2.0#changePassword</operationID>
</operationIdentifier>
```

- `resetPassword`: Reset a user's password (for delegated use only):

```
<operationIdentifier operationIdType='urn:oasis:names:tc:SPML:1:0#URN'>
   <operationID>urn:trulogica:concero:2.0#resetPassword</operationID>
</operationIdentifier>
```

- `enable`: enable a user (and any secondary users, if this is a primary user):

```
<operationIdentifier operationIdType='urn:oasis:names:tc:SPML:1:0#URN'>
   <operationID>urn:trulogica:concero:2.0#enable</operationID>
</operationIdentifier>
```

- `disable`: disable a user (and any secondary users, if this is a primary user):

```
<operationIdentifier operationIdType='urn:oasis:names:tc:SPML:1:0#URN'>
   <operationID>urn:trulogica:concero:2.0#disable</operationID>
</operationIdentifier>
```

- `terminate`: terminate a user:

```
<operationIdentifier operationIdType='urn:oasis:names:tc:SPML:1:0#URN'>
   <operationID>urn:trulogica:concero:2.0#terminate</operationID>
</operationIdentifier>
```

The attributes for the request follow the `<operationIdentifier>` element:

```
<attributes>
   <attr name='urn:trulogica:concero:2.0#changePassword'>
      <value>passwd</value>
   </attr>
```

The new password for the user specified by the `<identifier>` element.

```
<attributes>
   <attr name='urn:trulogica:concero:2.0#resourceId'>
      <value>resource</value>
   </attr>
```

The resource on which to change the user's password.

Alternatively, to change a user's password on several specified resources, the following `<attributes>` can be used:

```
<attributes>
   <attr name='urn:trulogica:concero:2.0#rcPassword'>
      <value>attr:pwd</value>
      <value>new:lmn</value>
      <value>ABLDAP70</value>
      <value>ABLDAP72</value>
   </attr>
</attributes>
```

The four `<value>` items are as follows:

- Current password (`attr:pwd`)
- New password, (`new:lmn`)
- Two resources on which the password change is to take effect (`ABLDAP70` and `ABLDAP72`).

Enabling Service Membership

```
<extendedRequest requestID=12345
execution='urn:oasis:names:tc:SPML:1:0#asynchronous'>
   <operationalAttributes>
      <attr name='urn:oasis:names:tc:SPML:1:0#UserIDAndOrDomainName'>
         <value>admin_username</value>
      </attr>
      <attr name='urn:trulogica:concero:2.0#password'>
         <value>admin_passwd</value>
      </attr>
      <attr name='urn:trulogica:concero:2.0#serviceName'>
         <value>service_name</value>
      </attr>
   </operationalAttributes>

   <identifier type='urn:oasis:names:tc:SPML:
   1:0#UserIDAndOrDomainName'>
      <id>id</id>
   </identifier>
   <providerIdentifier providerIDType='urn:oasis:names:tc:SPML:
   1:0#URN'>
      <providerID>urn:trulogica:concero:2.0</providerID>
   </providerIdentifier>
   <operationIdentifier operationIDType='urn:oasis:names:tc:SPML:
   1:0#URN'>
      <operationID>urn:trulogica:concero:2.0#enableMembership
      </operationID>
   </operationIdentifier>
   <attributes/>
</extendedRequest>
```

Disabling Service Membership

```
<extendedRequest requestID=value
execution='urn:oasis:names:tc:SPML:1:0#asynchronous'>
   <operationalAttributes>
      <attr name='urn:oasis:names:tc:SPML:1:0#UserIDAndOrDomainName'>
         <value>admin_username</value>
      </attr>
      <attr name='urn:trulogica:concero:2.0#password'>
         <value>admin_passwd</value>
      </attr>
      <attr name='urn:trulogica:concero:2.0#serviceName'>
      </attr>
   </operationalAttributes>
   <identifier type='urn:oasis:names:tc:SPML:1:0#UserIDAndOrDomainName'>
      <id>id</id>
   </identifier>
   <providerIdentifier providerIDType='urn:oasis:names:tc:SPML:1:0#URN'>
      <providerID>urn:trulogica:concero:2.0</providerID>
   </providerIdentifier>
   <operationIdentifier operationIDType=
   'urn:oasis:names:tc:SPML:1:0#URN'>
      <operationID>urn:trulogica:concero:2.0#disableMembership
      </operationID>
   </operationIdentifier>
   <attributes/>
</extendedRequest>
```

Resetting a User's Password

```
<extendedRequest requestID=value execution='urn:oasis:names:tc:SPML:
1:0#asynchronous'>
   <operationalAttributes>
      <attr name='urn:oasis:names:tc:SPML:1:0#UserIDAndOrDomainName'>
         <value>admin_username</value>
      </attr>
      <attr name='urn:trulogica:concero:2.0#password'>
         <value>pw123</value>
      </attr>
   </operationalAttributes>

   <identifier type='urn:oasis:names:tc:SPML:1:0#UserIDAndOrDomainName'>
      <id>userID</id>
   </identifier>
   <providerIdentifier providerIDType='urn:oasis:names:tc:SPML:1:0#URN'>
      <providerID>urn:trulogica:concero:2.0</providerID>
   </providerIdentifier>
   <operationIdentifier operationIdType='urn:oasis:names:tc:
   SPML:1:0#URN'>
      <operationID>urn:trulogica:concero:2.0#resetPassword
      </operationID>
   </operationIdentifier>

   <attributes>
      <attr name='urn:trulogica:concero:2.0#rcPassword'>
         <value>new:abcd1234</value>
      </attr>
   </attributes>
</extendedRequest>
```

Enabling a User Account

```
<extendedRequest requestID=value execution='urn:oasis:names:tc:SPML:
1:0#asynchronous'>
   <operationalAttributes>
      <attr name='urn:oasis:names:tc:SPML:1:0#UserIDAndOrDomainName'>
         <value>admin_username</value>
      </attr>
      <attr name='urn:trulogica:concero:2.0#password'>
         <value>admin_passwd</value>
      </attr>
   </operationalAttributes>

   <identifier type='urn:oasis:names:tc:SPML:1:0#UserIDAndOrDomainName'>
      <id>id</id>
   </identifier>
   <providerIdentifier providerIDType='urn:oasis:names:tc:SPML:1:0#URN'>
      <providerID>urn:trulogica:concero:2.0</providerID>
   </providerIdentifier>
   <operationIdentifier operationIDType='urn:oasis:names:tc:SPML:
   1:0#URN'>
      <operationID>urn:trulogica:concero:2.0#enable</operationID>
   </operationIdentifier>
   <attributes/>
</extendedRequest>
```

Disabling a User Account

```
<extendedRequest requestID='1769' execution='urn:oasis:names:tc:SPML:
1:0#asynchronous'>
   <operationalAttributes>
      <attr name='urn:oasis:names:tc:SPML:1:0#UserIDAndOrDomainName'>
         <value>sisa</value>
      </attr>
      <attr name='urn:trulogica:concero:2.0#password'>
         <value>abc123</value>
      </attr>
   </operationalAttributes>

   <identifier type='urn:oasis:names:tc:SPML:1:0#UserIDAndOrDomainName'>
      <id>wsuser1</id>
   </identifier>
   <providerIdentifier providerIDType='urn:oasis:names:tc:SPML:1:0#URN'>
      <providerID>urn:trulogica:concero:2.0</providerID>
   </providerIdentifier>
   <operationIdentifier operationIDType='urn:oasis:names:tc:SPML:1:0#URN'>
      <operationID>urn:trulogica:concero:2.0#disable</operationID>
   </operationIdentifier>
   <attributes/>
</extendedRequest>
```

Terminating a User Account

```
<extendedRequest requestID=value execution='urn:oasis:names:tc:SPML:
1:0#asynchronous'>
   <operationalAttributes>
      <attr name='urn:oasis:names:tc:SPML:1:0#UserIDAndOrDomainName'>
         <value>admin_username</value>
      </attr>
      <attr name='urn:trulogica:concero:2.0#password'>
```

```
            <value>admin_passwd</value>
         </attr>
      </operationalAttributes>

      <identifier type='urn:oasis:names:tc:SPML:1:0#UserIDAndOrDomainName'>
         <id>id</id>
      </identifier>
      <providerIdentifier providerIDType='urn:oasis:names:tc:SPML:1:0#URN'>
         <providerID>urn:trulogica:concero:2.0</providerID>
      </providerIdentifier>
      <operationIdentifier operationIDType='urn:oasis:names:tc:
      SPML:1:0#URN'>
         <operationID>urn:trulogica:concero:2.0#terminate</operationID>
      </operationIdentifier>
   </extendedRequest>
```

<modifyRequest>

This request type is used to modify an existing user account.

You cannot modify a user's password using the `<modifyRequest>` element; use the `<extendedRequest>` element with the `passwordReset` operational identifier for this purpose. If you modify a `Password` attribute, this causes the appropriate password request to be generated internally.

The `<modifyRequest>` element cannot be used to add or remove service subscriptions. These operations are also performed using `<addRequest>` and `<deleteRequest>`.

If you use the `Replace` operation in a `<modifyRequest>` to modify an entitlements attribute, ensure that the user's current entitlements are specified as part of the request. If you do not specify the existing entitlements, they are removed from the user.

Elements and Attributes

```
<modifyRequest ... requestID=value>
```

The ID of the provisioning transaction. This ID is present in the `<requestResponse>`.

```
<operationalAttributes>
   <attr name='urn:oasis:names:tc:SPML:1:0#UserIDAndOrDomainName'>
      <value>username</value>
   </attr>
   <attr name='urn:trulogica:concero:2.0#password'>
      <value>123passwd</value>
   </attr>
```

The user name and password of the administrator authorized to add a user to the service, or the user if this is a self-service request. Select Identity authenticates this user against the service. Specify this element for simple modify requests or modify requests issued during reconciliation from an authoritative or non-authoritative resource.

```
   <operationalAttributes>
      <attr name='urn:trulogica:concero:2.0#serviceName'>
         <value>service_name</value>
      </attr>
```

The name of the service where the user to be modified is a member. Specify this attribute for user provisioning (from Select Identity to the resource) in modify requests only. Omit this attribute from modify profile requests.

```
<operationalAttributes>
   <attr name='urn:trulogica:concero:2.0#reverseSync'>
      <value>true</value>
   </attr>
```

Indicates that the operation is a reverse synchronization (data is pushed from the resource to Select Identity). This value must always be true. Include this attribute in reverse synchronization or in modify requests issued during reconciliation by an authoritative or non-authoritative resource.

```
<operationalAttributes>
   <attr name='urn:trulogica:concero:2.0#resourceId'>
      <value>resource_ID</value>
   </attr>
```

The resource issuing the request, as in the case of auto-discovery or synchronization of resource data. Include this attribute for reverse synchronization or in modify requests issued during reconciliation by an authoritative or non-authoritative resource.

```
<operationalAttributes>
   <attr name='urn:trulogica:concero:2.0#keyFields'>
      <value>field</value>
   </attr>
```

The key fields to update in the Select Identity database during reconciliation. Include the `keyFields` attribute in modify requests issued during reconciliation by an authoritative or non-authoritative resource.

```
   <identifier ...>
      <id>value</id>
   </identifier>
```

The Select Identity ID of the user to modify. If this is the same as the user specified in the `<operationalAttributes>`, the request is self-service.

```
<modification name=attr_name operation=replace>
   <value>value</value>
</modification>
```

The name-value pair of the attribute to modify and the operation to perform, where `attr_name` is the name of the attribute and `value` is the value. Each attribute name is defined when an administrator creates a service in Select Identity.

If performing reverse synchronization, the attribute names are defined by the resource, and Select Identity determines this based on rules.

Available Operations

Available operations for the `<modifyRequest>` depend on whether the target attribute is single- or multi-value, and are as follows:

- `add`: If the target is a single-value attribute and is empty, this adds the specified value. If the target is a multi-value attribute, it adds the value to the existing list of values. If the target attribute is single-value and already populated, this overwrites the existing value.
- `delete`: If the target is a single-value attribute and is populated, this deletes the specified value (subject to an attribute named `AllowDeletionSetting`). If the target is single-value and empty, the operation does nothing. If the target is a multi-value attribute, it deletes all values by default.
- `deleteattr`: Deletes the attribute as well as the value on a single-value attribute. Deletes the attribute and all values on a multi-value attribute.

- `replace`: Replaces the value of an attribute with the value specified. It overwrites all values of a multi-value field with the specified value. If the attribute is single-value and empty, this operation adds the value to the attribute.

If you wish to modify entitlements, specify the resource name using the following format, and provide a value element for each attribute you wish to grant. Specify `add` or `delete` (instead of `replace`) as the operation. If performing reverse synchronization, you do not need to specify the `resource`:

```
<modification name='urn:trulogica:concero:2.0#groups:resourceName'
operation=add>
```

<searchRequest>

The following is a partial example of a `<searchRequest>` . Use this element to perform user account searches on various criteria. You can only search for user accounts; searches for other objects such as services or resources, are not currently supported.

Only non-sensitive attributes are retrieved. Sensitive (encrypted) attributes are not retrieved.

Web Services search operations are case-sensitive.

The behavior of this request type can be configured along with the user search functionality in the Select Identity Web interface. This is done by adding, removing, or changing columns in the **TAUser** table.

The **TAUser** table columns are controlled by a setting in the `TruAccess.properties` file, which is documented in an appendix to the *HP OpenView Select Identity Installation Guide*.

If you do not include <searchBase> or a `<filter>` attribute, a `<searchRequest>` returns all users. Therefore, use the TruAccess properties to set the maximum possible number of search results that an operation will return. You can also use the available `<filter>` attributes to set search criteria. The following are the default search criteria attributes:

- UserName
- Email
- FirstName
- LastName
- Status

The `Status` attribute contains the **User State Status** value. For more information about its possible values, refer to the *HP OpenView Select Identity Online Help for Administration*. An example `<filter>` attribute containing all of the possible values is provided below, with the textual value name in a comment line:

```
<filter>
   <equalityMatch name="urn:oasis:names:tc:SPML:1:0#status">
      <value>106</value>
         <!- Unlocked ->
      <value>101</value>
         <!- Created ->
      <value>100</value>
         <!- Enabled ->
     <value>10</value>
         <!- Ready ->
     <value>1</value>
         <!- Pending ->
      <value>0</value>
```

```
            <!- Waiting ->
          <value>-1</value>
            <!- Rejected ->
          <value>-10</value>
            <!- Disabled ->
          <value>-100</value>
            <!- Deleted ->
          <value>-101</value>
            <!- Terminated ->
          <value>-102</value>
            <!- Locked ->
        </equalityMatch>
      </filter>
```

The number of records retrieved by a `<searchRequest>` can be limited on a per-request basis by adding `maxResultSize` in an `equalityMatch` attribute, as follows:

```
<filter>
   <equalityMatch name='urn:trulogica:concero:2.0#maxResultSize'>
      <value>10</value>
   </equalityMatch>
</filter>
```

Elements and Attributes

```
<searchRequest ... requestID=12345>
```

The ID of the provisioning transaction. This ID is present in the <requestResponse>.

```
<operationalAttributes>
   <attr name='urn:oasis:names:tc:SPML:1:0#UserIDAndOrDomainName'>
      <value>username</value>
   </attr>
   <attr name='urn:trulogica:concero:2.0#password'>
      <value>admin_passwd</value>
   </attr>
```

The user name and password of the administrator authorized to search for a user on the service. Select Identity authenticates this administrative user against the service.

```
<attr name='urn:hp:selectidentity:4.0#getPermissions' >
   <dsml:value>true</dsml:value>
</attr>
```

This optional attribute specifies a search operation type. In this case, `getPermissions`, which retrieves permissions for the specified user.

```
<searchBase type='urn:oasis:names:tc:SPML:1:0#UserIDAndOrDomainName'>
   <id>user_ID</id>
</searchBase>
```

This optional attribute specifies the user whose attributes you want to retrieve from Select Identity.

```
<filter>
   <equalityMatch name=;urn:trulogica:concero:2.0#resourceId'>
      <value>resourceName</value>
   </equalityMatch>
</filter>
```

Specifies a search filter, which in this case retrieves matches on a specific resource (`resourceName`). You can also use `serviceName`.

```
<attributes>
   <attr name='ConceroUserId'/>
   <attr name='ConceroEmail'/>
```

```
        <attr name='FirstName'/>
        <attr name='LastName'/>
    </attributes>
```

Attributes used for search criteria. If you change the TruAccess property that controls the available search attributes, this portion of the request should use attributes that match this setting.

Search Results

Search results are returned within a `<searchResponse>` element.

Self-Service Requests

Web Services requests can be used for self-service operations. In a self-service request, the Requestor and the target user are the same whereas a delegated request contains an administrator's username and password as the originator. In most other respects, self service and delegated requests are closely similar.

Users who originate their own `<addRequests>` for secondary user IDs must have the Add Account permission in their role on the service where the account is to be added.