

# **HP OpenView Select Identity**

**Connector for  
SAP Release 3, Version 4.7**

## **Installation and Configuration Guide**

**Connector Version: 3.5  
Select Identity Version: 3.3.1**



**August 2005**

© 2005 Hewlett-Packard Development Company, L.P.

## 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

© Copyright 2005 Hewlett-Packard Development Company, L.P.

No part of this document may be copied, reproduced, or translated into another language without the prior written consent of Hewlett-Packard Company. The information contained in this material is subject to change without notice.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>). Portions Copyright (c) 1999-2003 The Apache Software Foundation. All rights reserved.

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.

This product includes software developed by Teodor Danciu <http://jasperreports.sourceforge.net>). Portions Copyright (C) 2001-2004 Teodor Danciu (teodord@users.sourceforge.net). All rights reserved.

Portions Copyright 1994-2004 Sun Microsystems, Inc. All Rights Reserved.

This product includes software developed by the Waveset Technologies, Inc. ([www.waveset.com](http://www.waveset.com)). Portions Copyright © 2003 Waveset Technologies, Inc. 6034 West Courtyard Drive, Suite 210, Austin, Texas 78730. All rights reserved.

Portions Copyright (c) 2001-2004, Gaudenz Alder. All rights reserved.

## Trademark Notices

HP OpenView Select Identity is a trademark of Hewlett-Packard Development Company, L.P. Microsoft, Windows, the Windows logo, and SQL Server are trademarks or registered trademarks of Microsoft Corporation.

Sun™ workstation, Solaris Operating Environment™ software, SPARCstation™ 20 system, Java technology, and Sun RPC are registered trademarks or trademarks of Sun Microsystems, Inc. JavaScript is a trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

This product includes the Sun Java Runtime. This product includes code licensed from RSA Security, Inc. Some portions licensed from IBM are available at <http://oss.software.ibm.com/icu4j/>.

IBM, DB2 Universal Database, DB2, WebSphere, and the IBM logo are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group.

This product includes software provided by the World Wide Web Consortium. This software includes xml-apis. Copyright © 1994-2000 World Wide Web Consortium, (Massachusetts Institute of Technology, Institute National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved. <http://www.w3.org/Consortium/Legal/>

Intel and Pentium are trademarks or registered trademarks of Intel Corporation in the United States, other countries, or both.

AMD and the AMD logo are trademarks of Advanced Micro Devices, Inc.

BEA and WebLogic are registered trademarks of BEA Systems, Inc.

VeriSign is a registered trademark of VeriSign, Inc. Copyright © 2001 VeriSign, Inc. All rights reserved.

All other product names are the property of their respective trademark or service mark holders and are hereby acknowledged.

## Support

Please visit the HP OpenView web site at:

**<http://www.managementsoftware.hp.com/>**

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

You can also go directly to the support web site at:

**<http://support.openview.hp.com/>**

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 valuable support customer, you can benefit by using the support site to:

- Search for knowledge documents of interest
- 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://support.openview.hp.com/access\\_level.jsp](http://support.openview.hp.com/access_level.jsp)**

To register for an HP Passport ID, go to:

**<https://passport2.hp.com/hpp/newuser.do>**

# contents

<b>Chapter 1</b>	<b>Installing the Connector</b> .....	7
	System Requirements. ....	8
	Deploying on the Application Server. ....	9
	Installing the Agent .....	10
	Prerequisites .....	11
	Deployment .....	11
	Agent Configuration .....	13
	Understanding the Configuration Files .....	13
	Understanding the Stylesheet. ....	17
<b>Chapter 2</b>	<b>Understanding the Mapping File</b> .....	23
	General Information. ....	24
	SAP Mapping Information. ....	28
<b>Chapter 3</b>	<b>Configuring the Connector</b> .....	30
<b>Chapter 4</b>	<b>Uninstalling the Connector</b> .....	33
	Uninstalling the Connector from WebLogic .....	33
	Uninstalling the Agent. ....	34

# Installing the Connector

The SAP connector enables HP OpenView Select Identity to provision SAP users on target SAP R/3 systems. The following operations are supported by the connector in this mode of operation:

- Add, update, and remove users
- Retrieve user attributes
- Enable and disable users
- Verify a user's existence
- Change user passwords
- Reset user passwords
- Retrieve all entitlements (profiles and roles)
- Retrieve a list of supported user attributes
- Assign and unassign entitlements (profiles and roles) to and from users

The connector also enables the SAP resource to reconcile SAP HR Employee data with Select Identity; it can push changes made in SAP back to the Select Identity server. To enable this functionality, you must install the SAP agent. If the following events occur on SAP resource, the changes can be reconciled with Select Identity:

- Add new employee

- Modify employee
- Terminate employee

These modes of operation (user provisioning and reconciliation) are referred to in this guide as SAP User mode and SAP HR Employee mode, respectively. The configuration of the connector for both modes is in this guide.

The SAP connector is packaged in the following files:

- `sapr3schema.jar` — contains the attribute mapping files for this system
- `sapr3connector.rar` — contains the connector binary files
- `SAPHRAgent.zip` — contains the agent binary and configuration files

These files are in the SAP directory on the Select Identity Connector CD.

## System Requirements

The SAP connector is supported in the following environment:

Select Identity Version	Application Server	Database
3.0.2	WebLogic 8.1.2 on Windows 2003	SQL Server 2000
3.3	WebLogic 8.1.4 on Windows 2003	SQL Server 2000
3.3.1	WebLogic 8.1.4 on Windows 2003	SQL Server 2000

This connector is supported with SAP release 3 version 4.7 on HP-UX 11i.

In addition, this connector requires the SAP Java Connector (JCo) API 2.1.3 to run on the web application server. Install this API before installing and running the SAP connector. Distribution packages are available for various JRE versions and hardware processors. Log on to <http://service.sap.com> and download the appropriate file from the Downloads section:

- `sapjco-ntintel-2.1.3.zip` for a 32-bit JRE running on a 32-bit INTEL x86 or a 64-bit INTEL Itanium processor
- `sapjco-ntia64-2.1.3.zip` for a 64-bit JRE running on a 64-bit INTEL Itanium processor



Perform the following to install the SAP JCo API:

- 1 Unzip the distribution package in a designated directory on the server.
- 2 If you have an existing `librfc32.dll` in the `System32` directory, replace it with the one that comes with the JCo API. Also, copy the `sapjcorfc.dll` file that comes with the JCo API to the `System32` directory.
- 3 Add the SAP JCo installation path to the `PATH` environment variable.
- 4 Add `sapjco-install-path\sapjco.jar` to your `CLASSPATH` environment variable.

## Deploying on the Application Server

To install the SAP connector on the Select Identity server, complete these steps:

- 1 Create a subdirectory in the Select Identity home directory where the connector's RAR file will reside. For example, you could create the `C:\Select_Identity\connectors` folder on Windows. (A connector subdirectory may already exist.)
- 2 Copy the `sapr3connector.rar` file from the Select Identity Connector CD to the connector subdirectory.
- 3 Create a schema subdirectory in the Select Identity home directory where the connector's mapping file(s) will reside. For example, you could create the `C:\Select_Identity\schema` folder. (This subdirectory may exist.)
- 4 Extract the contents of the `sapr3schema.jar` file (on the Select Identity Connector CD) to the schema subdirectory.
- 5 Ensure that the `CLASSPATH` environment variable in the WebLogic server startup script references the schema subdirectory.
- 6 Start the application server if it is not currently running.
- 7 Log on to the WebLogic Server Console.
- 8 Navigate to *My\_domain* → **Deployments** → **Connector Modules**.
- 9 Click **Deploy a New Connector Module**.

- 10 Locate and select the `sapr3connector.rar` file from the list. It is stored in the connector subdirectory.
- 11 Click **Target Module**.
- 12 Select the **My Server** (your server instance) check box.
- 13 Click **Continue**. Review your settings.
- 14 Keep all default settings and click **Deploy**. The Status of Last Action column should display Success.
- 15 Modify the mapping file, if necessary. See [Understanding the Mapping File on page 23](#) for details.

After installing the connector, see [Configuring the Connector on page 30](#) to register and configure the connector in Select Identity.

## Installing the Agent

To enable the SAP connector to reconcile SAP HR Employee data changes made on the resource with the Select Identity server, you must install the SAP agent. The agent provides autodiscovery and reconciliation of SAP R/3 data; it identifies changes on the SAP system and sends this information to the Select Identity system, updating the corresponding identity information.

The SAP agent is generic and can be used on any SAP R/3 system (locally or on the network). It provides the flexibility to process all data entries of SAP JCo tables specified in the configuration files included with the agent.

The SAP agent is packaged in `SAPHRAgent.zip` file. This file contains the following files:

- `SAPHRAgent.jar` — contains the agent binaries
- `resources\sapHrConnector.properties` — specifies the agent properties including the location of the following configuration files:
  - `resources\startSapHrConnector.cmd` - command file to start the SAP agent
  - `xml\sapHrConnector.xml` — connectivity configuration to SAP R/3 and the Select Identity server

- `xml\sapHrConnectorBapiList.xml` — **BAPIs for which data is retrieved**
- `xml\sapHrConnectorStatus.xml` — **used internally by the agent**

## Prerequisites

The SAP agent requires the SAP Java Connector (JCo) API, version 2.1.2 or higher. You must install this API before installing and running the agent. To download the API, log on to **<http://service.sap.com>**. The API is available in the **Downloads** section. (If you installed the JCo API while installing the connector, you can skip this step.)

The implementation of the SAP agent relies on the following Java software components from the Apache Jakarta Project:

- Commons-logging
- Commons-httpclient
- Logging Services (log4j)

Additional third-party software used by Select Identity includes the following:

- 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
- OpenSPML Toolkit from OpenSPML.org

These libraries are shipped with the agent, in the `lib` folder.

## Deployment

The SAP agent is a stand-alone program; you do not have to install it on the SAP system. . The SAP agent is a stand-alone component and can be deployed in any of the following locations:

- On the Select Identity server
- On the SAP HR server

- On any other system that communicates freely with the SAP and Select Identity servers

No matter where the agent is installed, ensure that the SAP JCo libraries are installed correctly and the `sapjco.jar` is in the system class path.

After you choose where to deploy the agent (locally or remotely), perform the following steps to install the SAP agent:

- 1 Create a directory for the agent, which is referred to as the SAP agent directory in the remainder of this guide.
- 2 Extract the contents of the `SAPHRAgent.zip` file into this directory.
- 3 Ensure that the `JAVACLASSPATH` variable contains the location of the SAP JCo API `sapjco.jar` file, which resides in the SAP agent start script (`resources/startSapHrConnector.cmd`).
- 4 Identify the IP address, credentials, and clients of the SAP R/3 system.
- 5 Identify the IP address, port, service name, resource name, and Select Identity administrative user credentials.
- 6 Modify the `sapHrConnector.xml` file with this information as described in [Understanding the Configuration Files on page 13](#).
- 7 Identify the SAP business module and corresponding BAPI RFCs to be called, including the required parameters.
- 8 Modify the `sapHrConnectorBapiList.xml` and `sapHrConnectorStatus.xml` files, which are described in [Agent Configuration on page 13](#).
- 9 Copy and update the sample stylesheet to implement the data mapping. The stylesheet is described in [Understanding the Stylesheet on page 17](#).
- 10 If the SAP agent is planned to be executed within a scheduled task of the operating system, perform the corresponding steps.

Note that the Select Identity connectivity parameters, including the configured Service, must be available in Select Identity. See the *HP OpenView Select Identity Administrator Guide* for details on configuring resources and Services.

After completing these steps, the SAP agent is installed and ready to be executed. If it will run as a stand-alone program; the polling interval will trigger reconciling SAP data with Select Identity.

## Agent Configuration

The SAP agent is deployed with the following configuration files, which must be modified in order to configure the agent:

- `resources\sapHrConnector.properties` — specifies the agent properties including the location of the configuration files
- `xml\sapHrConnector.xml` — contains the connectivity configuration to SAP R/3 and the Select Identity server
- `xml\sapHrConnectorBapiList.xml` — describes the BAPIs for which data is retrieved

The mapping of resource data from the SAP system to Select Identity is configurable in the XSL stylesheet.

The following sections describe the configuration of the SAP agent in detail.

### Understanding the Configuration Files

The configuration files for the SAP agent contain information about connectivity to the SAP R/3 system, the BAPI RFCs to be performed to retrieve the data, and connectivity information for the Select Identity server to send SPML requests.



See the `SAP/SampleFiles` directory on the Select Identity Connector CD for an example of each configuration file.

#### `sapHrConnector.properties`

This properties file must be placed in the Java classpath. After installation, it is located in the *SAP agent directory/resources* directory and describes the following basic configuration parameters:

- `poll.interval` — Polling interval (in milliseconds) that specifies the scheduled data retrieval from SAP R/3 for reconciliation
- `sap.config.file` — Name and location of the configuration file containing the connectivity parameters for the SAP R/3 system and the Select Identity server
- `sap.bapi.config.file` — Name and location of the configuration file that specifies the BAPI RFCs called to retrieve the SAP R/3 data
- `status.file` — Internal status file, which should not be edited or moved

## sapHrConnector.xml

This configuration file describes the parameters for the agent to connect to the SAP R/3 system and the Select Identity server. After installation, it is located in the *SAP agent directory/xml* directory. Its location and name are configured in the `sapHrConnector.properties` file.

The XML configuration includes the following elements:

- `sapConnectionList` — The root element. There can be multiple connections, such as to support multiple SAP clients or users.
  - `sapConnection` — The element describing the connectivity for the SAP R/3 system including the following:
    - `sapApplicationServer` — IP address
    - `client` — SAP client
    - `language` — Language for the SAP connection
    - `sapSystem` — SAP system identifier
    - `user` — User name for authentication to SAP R/3
    - `password` — Corresponding password for the authentication
  - `ovsiConfiguration` — The element describing the configuration for the Select Identity SPML client component
    - `ovsiConnection` — Connectivity parameters for the Select Identity server including the following:
      - `ovsiRequestUrl` — IP address, port, and URI of the SOAP web server.
      - `ovsiAdminName` — Select Identity administrator name allowed to perform the SPML requests.
      - `ovsiAdminPwd` — Password of the Select Identity administrator. To encrypt the password, run `encode.bat` (on Windows) or `encode.sh` (on UNIX), which is provided in the `weblogic/keystore` subdirectory in the Select Identity home directory. This utility prompts you for the password to encrypt and will generate

the encrypted password. Be sure to copy the entire encrypted password in the field, as shown here:

```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\Documents and Settings\gadiap>cd C:\si3.3.1\weblogic\keystore
C:\si3.3.1\weblogic\keystore>encode.bat
Please enter the string to encode :abc123
Please enter the string to encode again :abc123
<ENC:1:cfrM1Gvylj+a8XEWiWo5cx9A+PC0WeS2Z0LieW0dUheR/jGrgha54K0k060hlmrvND2tRuzjo
GuVeEEDBpUmWo2dTN11ywhxwEDnsZLFxT4r349W/O/6sgoPbuJt3C4vYs8rQk0KpeUnq21G9bf tJbuU0
Bjyk6vU5qCTS?Rr1Ds=>
C:\si3.3.1\weblogic\keystore>

```

- `ovsiTargetDir` — Target location to which the SPML request files are written (only required if the Select Identity Web Service is not used; for file-based reconciliation).
- `maxMessages` — Maximum number of messages.

The `ovsiConnection` element is required in order to send SPML requests over the network to the Select Identity SOAP Web Service. If there are requirements that do not allow sending requests over the network, the SAP agent supports writing SPML files to the directory specified by the `ovsiTargetDir` parameter. These can then be used for file-based reconciliation with Select Identity.

The `maxMessages` parameter specifies the maximum number of SPML messages contained in one of these reconciliation files. This parameter may be used for tuning file-based reconciliation.

### sapHrConnectorBapiList.xml

This configuration file describes the BAPI RFCs called through the SAP JCo API to retrieve the data from the SAP R/3 system. After installation, it is located in the *SAP agent directory/xml* directory. Its location and name are configured in the `sapHrConnector.properties` file.

The XML configuration includes the following elements:

- `bapiList` — The root element supporting the data retrieval using multiple BAPIs

- `bapi` — The element specifying the BAPI RFC and describing the following detailed parameters:
  - `importParms` — Import (input) parameters for the BAPI call already filtering the results on the specified condition
    - `parm` — Single import parameter element
    - `field` — Field name of the import parameter
    - `value` — Value of the import parameter
  - `tables` — The element describing the list of SAP export tables to be read from the BAPI call
    - `table` — Single table element containing the `name` (SAP name of the table) and `key` (unique key field of the table that is used to equi-join multiple tables specified for a BAPI) elements
- `createDateField` — Field name of the export table(s) containing the timestamp for creation of an entry
- `modDateField` — Field name of the export table(s) containing the timestamp for modification of an entry
- `deleteDateField` — Field name of the export table(s) containing the timestamp for deletion of an entry
- `addOperation` — Flag specifying whether SPML `addRequests` are submitted for new entries
- `modifyOperation` — Flag specifying whether SPML `modifyRequests` are submitted for modified entries
- `deleteOperation` — Flag specifying whether SPML `deleteRequests` are submitted for outdated entries
- `ovsiResourceName` — Select Identity resource name required for reconciliation
- `ovsiServiceName` — Select Identity Service name for which delegated administrative requests are submitted
- `spmlStyleSheet` — Name and location of XSL stylesheet implementing the data mapping (see [Understanding the Stylesheet on page 17](#))



The SAP agent detects changes and change types based on configurable date fields of the corresponding SAP source table(s). These are compared with the last synchronization timestamp. The last synchronization timestamp is internally managed by the agent and is set after SPML requests are successfully created.

The date field name parameters are optional and they support initial loads with a specific request type, such as `<modifyOperation>true</modifyOperation>` and no date fields configured results in `modifyRequests` for all entries returned by the BAPI call.

### **sapHrConnectorStatus.xml**

This configuration file is virtually identical to the `sapHrConnectorBapiList.xml` file. It describes internally used parameters for each BAPI. After installation, it is located in the *SAP agent directory/xml* directory. Its location and name are configured in the `sapHrConnector.properties` file.

The XML configuration includes the following elements:

- `bapiList` — The root element supporting the data retrieval using multiple BAPIs
  - `bapi` — The element for a single BAPI
    - `rfm` — Name of the BAPI RFC
    - `num` — Identifier for the BAPI used to internally reference `sapHrConnectorStatus.xml` and `sapHrConnectorBapiList.xml`
    - `syncTable` — Name of the export table resulting from the RFC
    - `sync` — Last synchronization timestamp

Note that this configuration file is only modified to add or remove BAPIs and export tables configured in `sapHrConnectorBapiList.xml`.

## **Understanding the Stylesheet**

The XSL stylesheet of the SAP agent describes the transformation of resource data into Select Identity SPML requests and implements the data mapping of resource attributes required by Select Identity. Data mapping is required because attribute names in SAP R/3 and other resources can be different from

attributes names used in Select Identity. Furthermore, XSL stylesheets provide the flexibility to implement filtering, data mapping, and projection rules to reconcile the resource data with Select Identity.

The sample XSL stylesheet implements this functionality for the SAP HR module and the Employee BAPI.



See the `SAP/SampleFiles` directory on the Select Identity Connector CD for an example stylesheet.

### SampleSapEmployee.xsl

The SAP agent is installed with the sample XSL stylesheet called `SampleSapEmployee.xsl`. After installation, the file is located in the `SAP agent directory/xml` directory.

The XSL stylesheet includes the following SPML request handlers. Note that additional filter and data projection rules may be implemented for the request handlers, such as for building concatenated attribute values or setting constant values.

Element Name	Description
BatchRequest Handler	Defines the elements and attributes to be written for the SPML BatchRequest component of reconciliation files written to the configured target directory. The main elements of a BatchRequest are the operational attributes and call the handlers for the AddRequest, ModifyRequest, and DeleteRequest elements of the BatchRequest.
AddRequest Handler	Defines the elements and attributes to be written for SPML AddRequests. The main elements of an SPML request are the operational attributes and user attributes. For each attribute, the Attribute Handler is called.

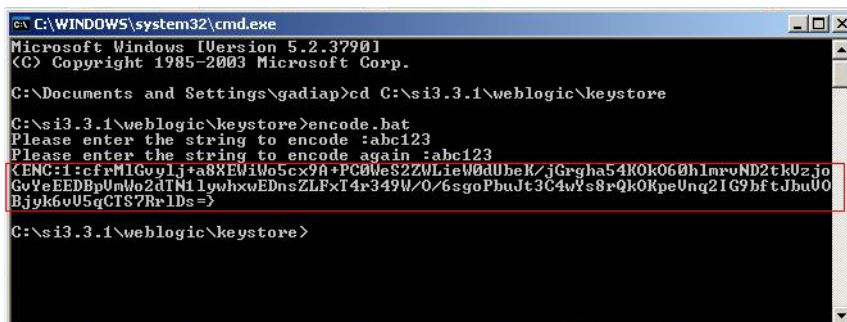
<b>Element Name</b>	<b>Description</b>
ModifyRequest Handler	Defines the elements and attributes to be written for SPML ModifyRequests. The main elements of an SPML request are the operational attributes, modifications to, and the identifier of the request target, which is an existing user in Select Identity. For each modification, the Attribute Handler is called.
DeleteRequest Handler	Defines the elements and attributes to be written for SPML DeleteRequests. The main elements of an SPML request are the operational attributes and the identifier of the request target. User attributes are not required for DeleteRequests.

As described above, each SPML request contains operational attributes. The operational attributes required by Select Identity are described in the following table:

<b>Attribute Name</b>	<b>Description</b>	<b>Type</b>
<b>Reconciliation Request</b>		
urn:trulogica:concerro:2.0#resourceId	The identifier of the Select Identity resource for which the reconciliation request is submitted.	AddModify-Delete
urn:trulogica:concerro:2.0#reverseSync	Whether the operation is a reverse synchronization. This value must always be true for reconciliation.	AddModify-Delete
urn:trulogica:concerro:2.0#keyFields	The key field based on which Select Identity field identifies the request target if not UserName.	ModifyDelete
urn:trulogica:concerro:2.0#taResourceKey	The attribute value of the resource key as configured in Select Identity.	ModifyDelete

<b>Attribute Name</b>	<b>Description</b>	<b>Type</b>
urn:trologica:concero:2.0#taUserName	For AddRequests, the attribute value of the resource key as configured in Select Identity.	Add
<b>Delegated Admin Requests</b>		
rn:trologica:concero:2.0#serviceName	The identifier of the Select Identity Service for which the delegated admin request is submitted.	AddModify-Delete
urn:oasis:names:tc:SPML:1:0#UserIDAndOrDomainName	Select Identity administrator name identifying the requestor.	AddModify-Delete
urn:trologica:concero:2.0#password	Select Identity administrator password for authenticating the requestor. This password should be generated using the encryption utility provided with Select Identity; see the note below this table for details.	AddModify-Delete

- ▶ To encrypt the password that is sent to the Select Identity server, run `encode.bat` (on Windows) or `encode.sh` (on UNIX), which is provided in the `weblogic/keystore` subdirectory in the Select Identity home directory. This utility prompts you for the password to encrypt and will generate the encrypted password. Be sure to copy the entire encrypted password in the field, as shown here:



```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\Documents and Settings\gadiap>cd C:\si3.3.1\weblogic\keystore

C:\si3.3.1\weblogic\keystore>encode.bat
Please enter the string to encode :abc123
Please enter the string to encode again :abc123
<ENC:1:cf:rM1Goy1j+a88EWiWo5cx9A+PC00eS2ZWLieW0dUbcK/jGrgha54R0k060h1mrvND2EkUzjo
GuYeEEDBpUmWo2dTN11ywhxwEDnsZLFxT4r349W/O/6sgoPhuJt3C4wYs8rQk0KpeUnq21G9bf tJbuU0
B.jyk6vU5qCTS7Rr1Ds=>

C:\si3.3.1\weblogic\keystore>

```

Reconciliation is the process of synchronizing resource data with Select Identity data. This means that in most cases, reconciliation requests are sent to the Select Identity Web Service. However, delegated admin requests may have to be submitted for specific requirements and if there is no Select Identity resource configured.

For details on the operational attributes of SPML requests, refer to the *HP OpenView Select Identity Web Service Developer Guide*.

### Attribute Mapping

One of the main components in the XSL stylesheet is the Attribute Handler called for each attribute or modification submitted for a request target. The attribute handler elements are defined by the following:

```

<!--
Handler to convert the attribute names
*****
-->
...
<xsl:when test="$ATTRNAME = '<resource attr name>'">
...
<xsl:attribute name="name">
<xsl:value-of select="'<OVSI attr name'" />

```

```
</xsl:attribute>  
...  
</xsl:when>
```

Attribute mappings can be added by performing the following steps:

- 1 Copy a single element.
- 2 Replace the resource attribute name (lower case).
- 3 Replace the Select Identity attribute name.

Note that all attributes configured as required in Select Identity must be part of the attribute mapping section in the stylesheet.

The attributes available for an SAP business object's export tables returned by the BAPI calls can be identified with the BAPI Browser of the SAP GUI or the SAP Interface Repository (<http://ifr.sap.com/catalog/query.asp>).

## Understanding the Mapping File

The SAP connector is deployed with the `Sap-R3.xml` mapping file, which describes the attributes required by the system. The file is created in XML, according to SPML standards, and is bundled in a JAR file called `sapr3schema.jar`. The mapping file is used to map user account additions and modifications from Select Identity to the system resource. When you deploy a resource using the Resources page of the Select Identity client, you can review this file.

You can create attributes that are specific to Select Identity using the Attributes page in the Select Identity client. These attributes can be used to associate Select Identity user accounts with system resources by editing the connector mapping file described in this chapter. This process becomes necessary because, for example, a single attribute “username” can have a different name on different resources, such as “login” for UNIX, “UID” for a database, and “userID” on a Windows server.

This file does not need to be edited unless you want to map additional attributes to your resource. If attributes and values are not defined in this mapping file, they cannot be saved to the resource through Select Identity.

## General Information

The following operations can be performed in the mapping file:

- Add a new attribute mapping
- Delete an existing attribute mapping
- Modify attribute mappings

Here is an explanation of the elements in the XML mapping files provided by the SAP connectors:

- **<Schema>**, **<providerID>**, and **<schemaID>**

Provides standard elements for header information.

- **<objectClassDefinition>**

Defines the actions that can be performed on the specified object as defined by that name attribute (in the **<properties>** element block) and the Select Identity-to-resource field mappings for the object (in the **<memberAttributes>** block). For example, the object class definition for users defines that users can be created, read, updated, deleted, reset, and expired in SAP.

- **<properties>**

Defines the operations that are supported on the object. This can be used to control the operations that are performed through Select Identity. The following operations can be controlled:

- Create (CREATE)
- Read (READ)
- Update (UPDATE)
- Delete (DELETE)
- Enable (ENABLE)
- Disable (DISABLE)
- Reset password (RESET\_PASSWORD)
- Expire password (EXPIRE\_PASSWORD)
- Change password (CHANGE\_PASSWORD)



The operation is assigned as the name of the <attr> element and access to the operation is assigned to a corresponding <value> element. You can set the values as follows:

- true — the operation is supported by the connector
- false — the operation is not supported by the connector
- bypass — the operation is not supported by the connector

Here is an example:

```
<objectClassDefinition name="User" description="SAP User">
  <properties>
    <attr name="CREATE">
      <value>true</value>
    </attr>
    <attr name="READ">
      <value>true</value>
    </attr>
```

- **<memberAttributes>**

Defines the attribute mappings. This element contains <attributeDefinitionReference> elements that describe the mapping for each attribute. Each <attributeDefinitionReference> must be followed by an <attributeDefinition> element that specifies details such as minimum length, maximum length, and so on.

Each <attributeDefinitionReference> element contains the following attributes:

- Name — the name of the reference.
- Required— if this attribute is required in the provisioning (set to true or false).
- Concero:tafield — the name of the Select Identity resource attribute. In general, the attribute assigned to tafield should be the same as the physical resource attribute, or at least the connector attribute. For example, it is recommended to have the following:

```
<attributeDefinitionReference name="FirstName"
  required="false" concero:tafield="[givenname]"
  concero:resfield="givenname" concero:init="true"
  concero:isMulti="true"/>
```

instead of this:

```
<attributeDefinitionReference name="FirstName"
required="false" concero:tafield="[FirstName]"
concero:resfield="givenname" concero:init="true"
concero:isMulti="true"/>
```

- **Concero:resfield** — the name of the physical resource attribute from the resource schema. If the resource does not support an explicit schema (such as UNIX), this can be a tag field that indicates a resource attribute mapping.

Also, the attribute name may be case-sensitive; for example, if the attribute is defined in all uppercase letters on the resource, be sure to specify it in all uppercase letters here.

- **Concero:isKey** — An optional attribute that, when set to true, specifies that this is the key field to identify the object on the resource. Only one `<attributeDefinitionReference>` can be specified where `isKey="true"`. This key field does not need to be the same as the key field of the identity object in Select Identity.

Note that for a key field mapping where `isKey="true"` and `tafield` is not assigned the `UserName` attribute, `UserName` should not be used in any other mapping. That is, `UserName` can be assigned to `tafield` only in cases where it is mapped to the key field in the resource. Example:

```
<attributeDefinitionReference name="UserName"
required="true" concero:tafield="[UserName]"
concero:resfield="uid" concero:isKey="true"
concero:init="true"/>
```

- **Concero:init** — An optional attribute that identifies that the attribute is initialized with the value of the attribute passed in from Select Identity.

Here is an example:

```
<memberAttributes>
<attributeDefinitionReference name="User Name"
required="true" concero:tafield="[User Name]"
concero:resfield="schema=newxj,table=T_USERINFO,
column=USERSTATUS" concero:isKey="true"
concero:init="true" />
```

The interpretation of the mapping between the connector field (as specified by the `Concero:tafield` attribute) and the resource field (as specified by the `Concero:resfield` attribute) is determined by the connector. The SAP connector has code to interpret the mappings in one way, as follows:

- The connector attribute names are specified in square braces, like this: `[xyz]`. The value of attribute `xyz` is taken from the UserModel during provisioning.
- Composite attributes can be specified in the SAP connector mapping file. To do this, specify `[attr1] xxxx [attr2]` as the connector attribute. This specifies that the value of the `attr1` and `attr2` attributes should be combined with the string `xxxx` to form a mapping for the specified resource field. SAP connector has code to handle these composite mappings.

- **<attributeDefinition>**

Defines the properties of each object's attribute. For example, the attribute definition for the Email attribute defines that it must be between zero and 100 characters in length and can contain the following letters, numbers, and characters: a-z, A-Z, 0-9, @, +, and a space.

Here is an excerpt from the `Sap-R3.xml` file:

```
<attributeDefinition name="Email" description="Email"
type="xsd:string" >
  <properties>
    <attr name="minLength">
      <value>0</value>
    </attr>
    <attr name="maxLength">
      <value>25</value>
    </attr>
    <attr name="pattern">
      <value><![CDATA[[a-zA-Z0-9@]+]]> </value>
    </attr>
  </properties>
</attributeDefinition>
```

- **<concero:entitlementMappingDefinition>**

Defines how entitlements are mapped to users.

- **<concero:objectStatus>**

Defines how to assign status to a user.

- **<concerno:relationshipDefinition>**  
Defines how to create relationships between users.

## SAP Mapping Information

The following are the attribute mappings supported for SAP systems. These are listed in the `Sap-R3.xml` mapping file. You can add, modify, or delete attributes once you are familiar with the contents of this file. You can edit the Select Identity resource attributes; they reflect the identity information as seen in Select Identity. The physical resource attributes are literal attributes of user accounts on the SAP server. These attributes cannot be changed. See the *HP OpenView Select Identity Connector Developer Guide* for more information about attributes and mapping information. .

Select Identity Resource Attribute	SAP Attribute	Description
Username	Username	Key field on the resource
Password	Password	
Firstname	Firstname	
Lastname	Lastname	
Middlename	Middlename	
Fullname	Fullname	
Department	Department	
City	City	
Country	Country	
Title	Title	
Email	E_Mail	
Zip	Postl_Cod1	
Address1	Building_P	
Address2	Floor_P	

<b>Select Identity Resource Attribute</b>	<b>SAP Attribute</b>	<b>Description</b>
Homephone	Tel1_Numbr	
Salutation	Title_P	
Costcenter	Kostl	
Company	Company	

## Configuring the Connector

After you deploy the connector on the application server, you must configure Select Identity to use the connector by deploying it in the Select Identity client. The following provides an overview of the procedures you must complete in order to deploy your connector. It also provides connector-specific information you must provide when configuring Select Identity to use the connector.

- 1 Register the connector with Select Identity by clicking the **Deploy New Connector** button on the Connectors home page. Complete this procedure as described in the “Connectors” chapter of the *HP OpenView Select Identity Administrator Guide*.

After you deploy the connector, the connector properties will look similar to this:

[Home](#) > [Connectors](#) : SAP

Connector Information	
* Connector Name:	SAP
* Pool Name:	eis/SAPR3

- 2 Deploy a resource that uses the newly created connector. On the Resources home page, click the **Deploy New Resource** button. When configuring the resource, refer to the following table for parameters specific to this connector:

Field Name	Sample Values	Description
Resource Name	sap_server	Name given to the resource.
Resource Type	SAP	The connector that was deployed in <a href="#">Step 1 on page 30</a> .
Authoritative Source	No	Whether this resource is a system that is considered to be the authoritative source for user data in your environment. You must specify <b>No</b> because the connector cannot synchronize account data with the Select Identity server.
Associate to Group	Selected	Whether the system uses the concept of groups. For this connector, select this option.
User Name	admin	Administrative username.
Password	password123	Account password.
Client ID	00	Server client ID.
Language	EN	Specified language for the system.
Host Address	server.company.com	The address of the server.
System Number	00	The system number.
Mapping File	Sap-R3.xml	Name of the resource mapping file.

Complete the steps in this procedure as described in the “Resources” chapter of the *HP OpenView Select Identity Administrator Guide*. After

you deploy the resource for the SAP connector, the Access Info page of the resource properties will look similar to this:

[Home](#) > [Resources](#) > [View Resource : SAP](#)

Resource Access Information	
* Resource Name:	SAP
* User Name:	sap*
* Password:	*****
* Client ID:	00
* Language:	EN
* Host Address:	15.139.88.155
* System Number:	00
* Mapping File:	Sap-R3.xml

- 3 Create attributes that link Select Identity to the connector. For each mapping in the connector's mapping file, create an attribute using the Attributes capability on the Select Identity client. Refer to the "Attributes" chapter in the *HP OpenView Select Identity Administrator Guide* for more information.

After you create attributes for the SAP resource, the list of resources looks similar to this:

Resource Attribute	MinLength	MaxLength	Mapped To	Authoritative
Class	0	10	class	<input type="checkbox"/>
Cost Center	0	10	CostCenter	<input type="checkbox"/>
Email	0	50	Email	<input type="checkbox"/>
First Name	0	50	FirstName	<input type="checkbox"/>
Last Name	0	50	LastName	<input type="checkbox"/>
Password	0	10	Password	<input type="checkbox"/>
sapr3_ENTITLEMENTS	1	255	sapr3_ENTITLEMENTS	<input checked="" type="checkbox"/>
sapr3_KEY	1	255	sapr3_KEY	<input checked="" type="checkbox"/>
User Name	0	10	(Select one)	<input type="checkbox"/>

- 4 Create a Service that will use the newly created resource. To do so, click the **Deploy New Service** button on the Services home page. Complete this procedure as described in "Services" of the *HP OpenView Select Identity Administrator Guide*. You will reference your new resource created in [Step 2](#) while creating this service.



## Uninstalling the Connector

If you need to uninstall a connector from Select Identity, make sure that the following are performed:

- All resource dependencies are removed.
- The connector is deleted using the Select Identity client Connectors pages.

## Uninstalling the Connector from WebLogic

Perform the following to delete a connector:

- 1 Log on to the WebLogic Server Console.
- 2 Navigate to *My\_Domain* → **Deployments** → **Connector Modules**.
- 3 Click the delete icon next to the connector that you want to uninstall.
- 4 Click **Yes** to confirm the deletion.
- 5 Click **Continue**.

# Uninstalling the Agent

To uninstall the SAP agent from the system, perform the following steps:

- 1 If the agent is executed within a scheduled task of the operating system, remove this task from the system. If otherwise the SAP R/3 Agent is running as a stand-alone process, stop this process.
- 2 Log files and reconciliation files may be required for certain purposes. If required, back up the files; the location of the files are specified in `log4j.properties` and `sapHrConnector.xml`.
- 3 Remove the SAP agent directory.

The SAP agent does not leave registry information on the system. If the configuration was changed, files must be manually deleted from corresponding locations.