HP OpenView Select Identity

Attribute Mapping Utility User's Guide

Software Version: 3.3.1

July 2005

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/). 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.

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

To register for an HP Passport ID, go to:

https://passport2.hp.com/hpp/newuser.do

contents

1

Introduction

The Attribute Mapping Utility enables you to create the XML and XSL mapping files required by the following connectors:

- DB2 connector
- SQL Server connector
- Oracle connector
- Sybase connector
- Informix connector

With other connectors that provision users in a resource, the resource typically defines a fixed set of attributes. Therefore, the mapping files are also fixed. For the connectors listed above, you can provision users and entitlements into a database. Because the database schema can be defined in a number of ways, the mapping files can also vary widely. The Attribute Mapping Utility gives you the needed flexibility to map table columns and stored procedures to Select Identity attributes.

After you map the attributes, you save the mappings, which generates an XML file, for “forward” mapping, and an XSL file, for use during reverse synchronization.

This guide describes how to access the Attribute Mapping Utility, provides an overview to the utility’s user interface, and describes how to define user and entitlements mappings.

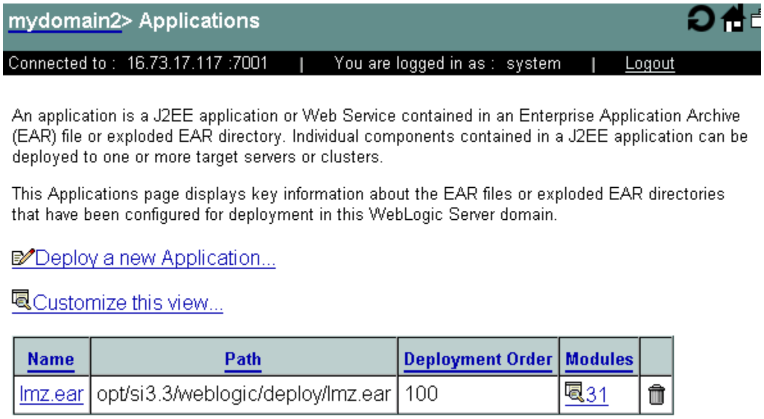
Deploying the Attribute Mapping Utility

When HP OpenView Select Identity is deployed on the application server, the WAR file for the Attribute Mapping Utility is also deployed. You do not need to perform any additional steps to deploy the utility.

To verify that the utility is deployed on WebLogic, perform the following steps:

1 Log on to the application server.

2 Navigate to ***My_domain*** →**Deployments** →**Application s**. A page similar to the following displays:



mydomain2> Applications

Connected to : 16.73.17.117 :7001 | You are logged in as : system | Logout

An application is a J2EE application or Web Service contained in an Enterprise Application Archive (EAR) file or exploded EAR directory. Individual components contained in a J2EE application can be deployed to one or more target servers or clusters.

This Applications page displays key information about the EAR files or exploded EAR directories that have been configured for deployment in this WebLogic Server domain.

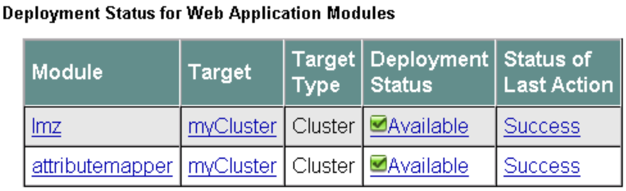
Deploy a new Application...

Customize this view...

Name	Path	Deployment Order	Modules	
lmz.ear	opt/si3.3/weblogic/deploy/lmz.ear	100	31	🗑

3 Click **lmz.ear**.

4 Click the **Deploy** tab. A page similar to the following displays; the attributemapper module should be listed at the bottom of the page:

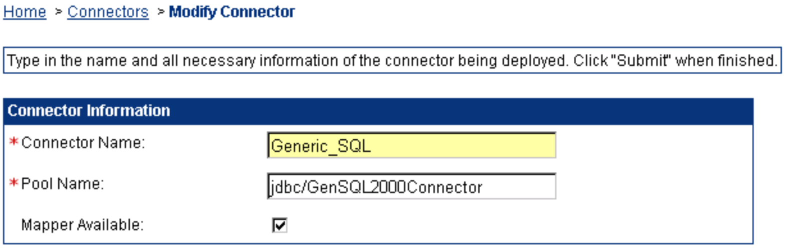


Deployment Status for Web Application Modules

Module	Target	Target Type	Deployment Status	Status of Last Action
lmz	myCluster	Cluster	✓Available	Success
attributemapper	myCluster	Cluster	✓Available	Success

Accessing the Attribute Mapping Utility

You can view the utility while creating or editing the connector's resource (using the Select Identity client) if the Mapper Available option was selected when the connector was deployed:



Home > Connectors > **Modify Connector**

Type in the name and all necessary information of the connector being deployed. Click "Submit" when finished.

Connector Information

*Connector Name:	Generic_SQL
*Pool Name:	jdbc/GenSQL2000Connector
Mapper Available:	☑

You can also view the utility as a standalone page.

To view the utility, perform one of the following procedures:

- Using the Select Identity client:

  a Click the **Resources** tab.

  b Click **Deploy New Resource**.

  c Provide the necessary values (refer to the connector's installation guide for values necessary to deploy the resource) and click the **Save & Continue** at the bottom of the resource's properties page until the Access Info page is displayed.

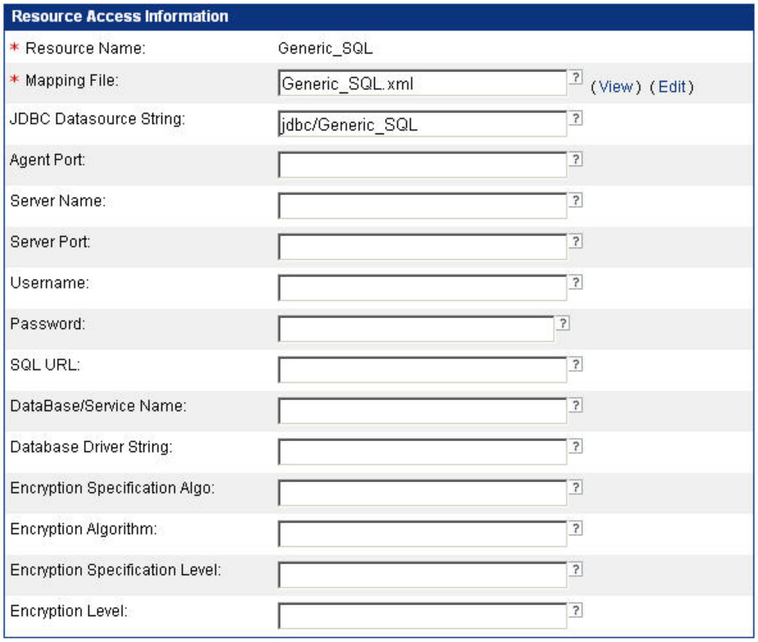
  d On the Access Info page, provide the necessary connection credentials, which vary depending on how the database connector and agent were installed and configured:

  — Using a JDBC data source, an agent is not installed:
  In this configuration, the connector performs operations on the database directly through JDBC calls. You must specify the JDBC data source and mapping file when configuring the resource.

  — Using a JDBC driver, an agent is not installed:
  The connector uses the JDBC driver to communicate with the database. You must specify all parameters except the agent port and JDBC data source.

— Using a JDBC driver, an agent installed:
If the agent is installed and a JDBC driver is used to communicate with the database, you must specify all parameters except the JDBC data source.

e Click the **Edit** link next to the Mapping File field.



Resource Access Information

* Resource Name:	Generic_SQL
* Mapping File:	Generic_SQL.xml (View) (Edit)
JDBC Datasource String:	jdbc/Generic_SQL
Agent Port:	
Server Name:	
Server Port:	
Username:	
Password:	
SQL URL:	
DataBase/Service Name:	
Database Driver String:	
Encryption Specification Algo:	
Encryption Algorithm:	
Encryption Specification Level:	
Encryption Level:	

To create or modify a mapping file for an existing resource, select the connector on the Resource tab and select **Modify Resource** from the Actions drop-down menu. View the Access Info page for the resource, and click the **Edit** link next to the Mapping field.

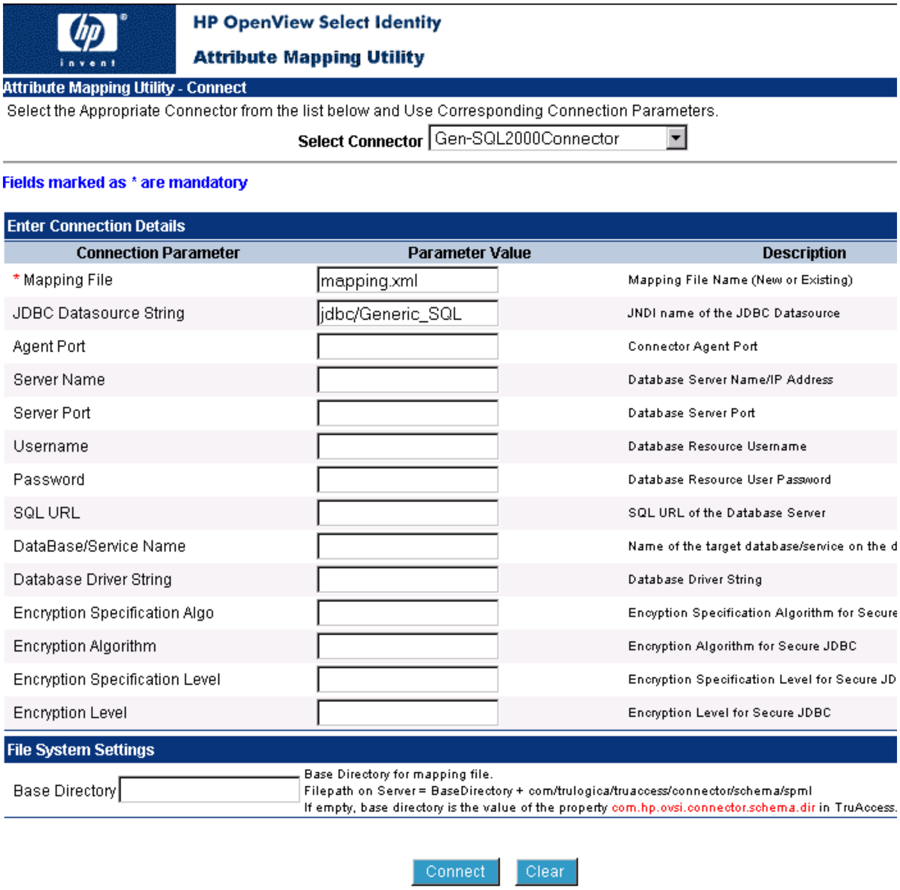
When you click the **Edit** link, the Attribute Mapping Utility displays and connects to the database using the values provided on the Access Info page. If you are creating a mapping file, a new XML file is created in the Select Identity home directory, in the `com/trulogica/truaccess/connector/schema/spml` subdirectory. (This default location can be configured by setting the `com.hp.ovsi.connector.schema.dir` parameter in the `TruAccess.properties` file.)

If the specified mapping file exists, the Attribute Mapping Utility displays, connects to the database, and load the existing settings in the mapping file.

- As a standalone page:
  Load the following URL in an Internet Explorer web browser (version 5.5+ with Javascript and Java support enabled):

  http://*hostname*:*port*/attributemapper/index.jsp

  where *hostname* is the name or IP address of the application server on which Select Identity is deployed and *port* is the application server's port. The following page displays:



HP OpenView Select Identity

Attribute Mapping Utility

Attribute Mapping Utility - Connect

Select the Appropriate Connector from the list below and Use Corresponding Connection Parameters.

Select Connector: Gen-SQL2000Connector

Fields marked as * are mandatory

Enter Connection Details

Connection Parameter	Parameter Value	Description
* Mapping File	mapping.xml	Mapping File Name (New or Existing)
JDBC Datasource String	jdbc/Generic_SQL	JNDI name of the JDBC Datasource
Agent Port		Connector Agent Port
Server Name		Database Server Name/IP Address
Server Port		Database Server Port
Username		Database Resource Username
Password		Database Resource User Password
SQL URL		SQL URL of the Database Server
DataBase/Service Name		Name of the target database/service on the d
Database Driver String		Database Driver String
Encryption Specification Algo		Encyption Specification Algorithm for Secure
Encryption Algorithm		Encryption Algorithm for Secure JDBC
Encryption Specification Level		Encryption Specification Level for Secure JD
Encryption Level		Encryption Level for Secure JDBC

File System Settings

Base Directory

Base Directory for mapping file.
Filepath on Server = BaseDirectory + com/trulogica/truaccess/connector/schema/spml
If empty, base directory is the value of the property com.hp.ovsi.connector.schema.dir in TruAccess.

To log in to the utility and connect to the database schema, which enables you to map tables to Select Identity attributes, complete the following steps:

a Select the connector for which you are mapping attributes from the Select Connector drop-down menu. This is the name of the connector deployed using the Select Identity client, on the Connectors tab. You can also select **None (Edit Mapping File Only)** from the drop-down list. This enables you to simply edit the XML mapping file on the Select Identity server, which then regenerates the XML and XSL files.

  The parameters you specify are determined by how the connector and agent were configured and installed, as described in Step d on page 10.

b *If you are using a JDBC data source to connect to the database:* Specify values in the following fields:

Field	**Value**
Mapping File	The name of the XML file that will be generated.
JDBC Datasource String	The JNDI name of the JDBC data source that was created on the Select Identity server to connect to the target database.

  Make sure all of the other fields are empty and go to Step f.

c *If you are using a JDBC driver to connect to the database but the agent is not installed:*
  Specify values in the following fields:

Field	**Value**
Mapping File	The name of the XML file that will be generated.
Server Name	The name of the database server.
Server Port	The database server’s listening port.
Username	The database user.
Password	The password of the specified user.

Field	Value
SQL URL	The name of the JDBC driver to use to connect to the database.
Database/Service Name	The name of the database.
Database Driver String	The JDBC driver being used.

d *If you are using a JDBC driver and the agent is installed:* Specify values in the following fields:

Field	Value
Mapping File	The name of the XML file that will be generated.
Server Name	The name of the database server.
Server Port	The database server's listening port.
Username	The database user.
Password	The password of the specified user.
Agent Port	The agent's listening port.
SQL URL	The name of the JDBC driver to use to connect to the database.
Database/Service Name	The name of the database.
Database Driver String	The JDBC driver being used.

e *If you select None (to edit the mapping file only):* Specify values in the following fields:

Field	Value
Resource Name	The name that will be stored in the XML file as the connector name.
Path to mapping file (on server)	The location and file name of the XML mapping file on the Select Identity server.

f In the Base Directory field, specify the top-level directory in which the XML file will reside after you map attributes. The `com/trulogica/truaccess/connector/schema/spml` directory structure is created in the specified location, and the XML and XSL files are stored created here.

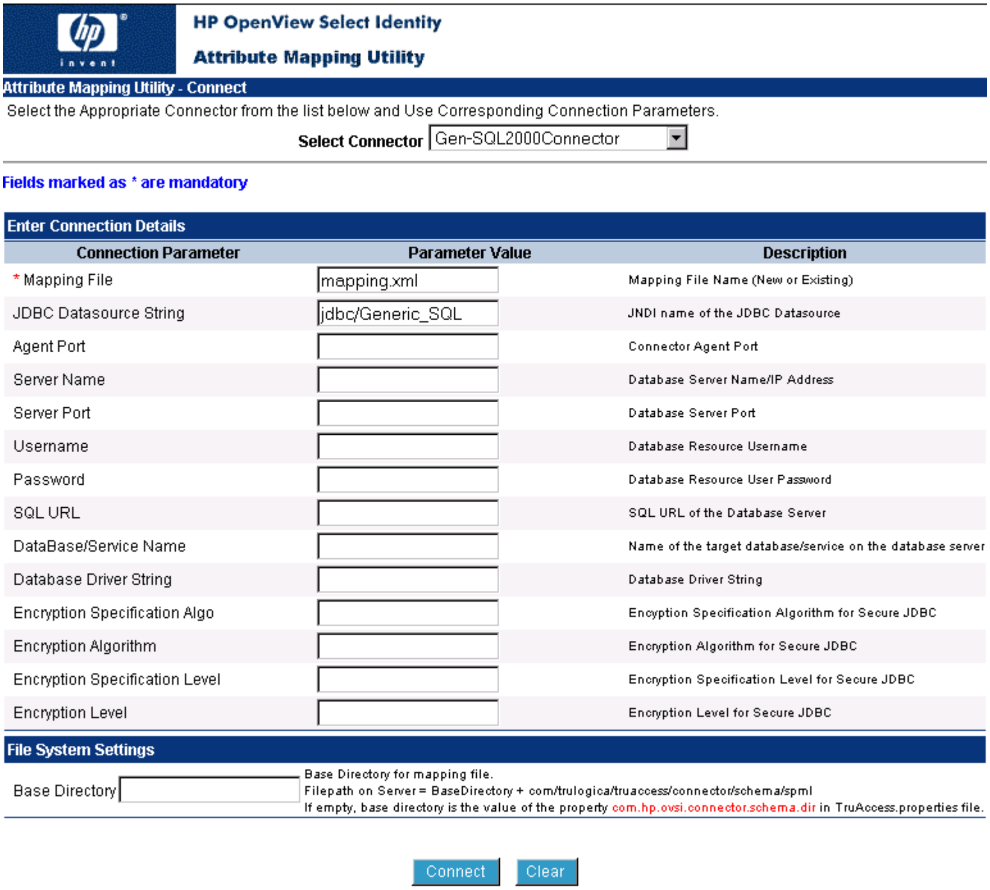
If the base directory is not specified, the value of the `com.hp.ovsi.connector.schema.dir` parameter in the `TruAccess.properties` file is used as the base directory. If neither of these values is provided, the files are stored in the application server's directory that is relevant to the current domain on the server.

g Click **Connect**.

If you could not log in to the Attribute Mapping Utility, verify the values specified in the fields. If you are using a JDBC data source, verify its name on the Select Identity server. These values were also specified when the connector’s resource was created using the Select Identity client.

Overview of the Utility

If you access the Attribute Mapping Utility by loading its URL in a browser, the first page that is displayed prompts you to enter connection information. These fields are explained on page 12.



HP OpenView Select Identity

Attribute Mapping Utility

Attribute Mapping Utility - Connect

Select the Appropriate Connector from the list below and Use Corresponding Connection Parameters.

Select Connector Gen-SQL2000Connector

Fields marked as * are mandatory

Enter Connection Details

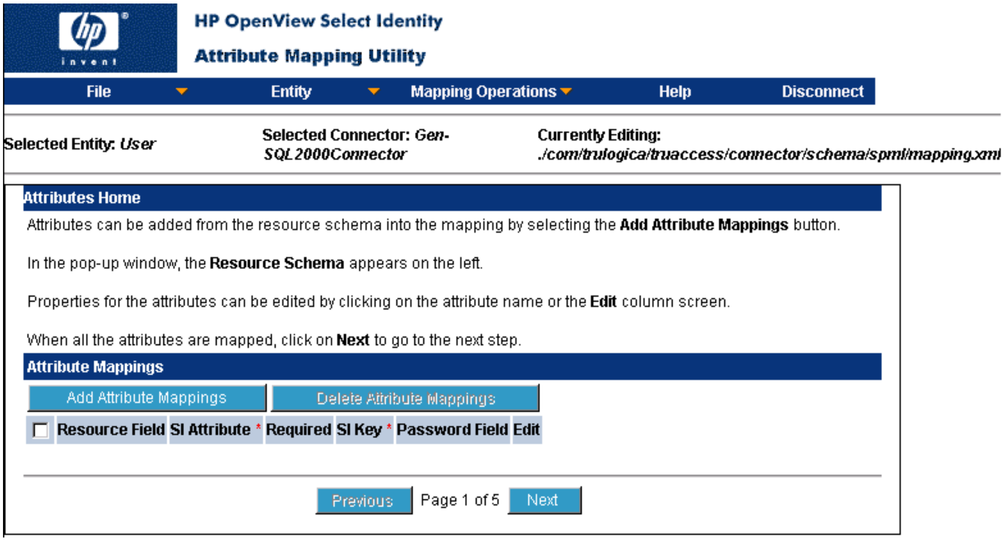
Connection Parameter	Parameter Value	Description
* Mapping File	mapping.xml	Mapping File Name (New or Existing)
JDBC Datasource String	jdbc/Generic_SQL	JNDI name of the JDBC Datasource
Agent Port		Connector Agent Port
Server Name		Database Server Name/IP Address
Server Port		Database Server Port
Username		Database Resource Username
Password		Database Resource User Password
SQL URL		SQL URL of the Database Server
DataBase/Service Name		Name of the target database/service on the database server
Database Driver String		Database Driver String
Encryption Specification Algo		Encyption Specification Algorithm for Secure JDBC
Encryption Algorithm		Encyption Algorithm for Secure JDBC
Encryption Specification Level		Encyption Specification Level for Secure JDBC
Encryption Level		Encyption Level for Secure JDBC

File System Settings

Base Directory

Base Directory for mapping file.
Filepath on Server = BaseDirectory + com/trulogica/truaccess/connector/schema/spml
If empty, base directory is the value of the property com.hp.ovsi.connector.schema.dir in TruAccess.properties file.

After you connect, or if you are accessing the Attribute Mapping Utility from the Resource page in the Select Identity client, the following page displays:



Selected Entity: *User*

Selected Connector: *Gen-SQL2000Connector*

Currently Editing: *./com/trulogica/truaccess/connector/schema/spml/mapping.xml*

Attributes Home

Attributes can be added from the resource schema into the mapping by selecting the **Add Attribute Mappings** button.

In the pop-up window, the **Resource Schema** appears on the left.

Properties for the attributes can be edited by clicking on the attribute name or the **Edit** column screen.

When all the attributes are mapped, click on **Next** to go to the next step.

Attribute Mappings

□	Resource Field	SI Attribute *	Required	SI Key *	Password Field	Edit

If you are editing an existing XML file, the mapped attributes are listed in the Attribute Mappings section of the page. If you are creating a new mapping file, the page appears as shown above.

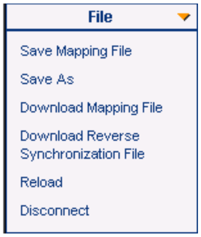
The Attribute Mapping Utility provides the following:

- Menus
- Mapping pages

Menus

The following menus and options are available.

- **File Menu**



Here are the options on this menu:

— **Save Mapping File**
Saves the XML and XSL file in the directory specified in the Base Directory field on the Select Identity server. If the base directory was not specified when logging in, the value of the `com.hp.ovsi.connector.schema.dir` property in the `TruAccess.properties` file is used. If neither of these values is available, the files are stored in the home directory of the application server.

— **Save As**
Enables you to save the XML and XSL files using a name other than that which was specified when you logged in to the utility. This displays pop-up dialog that prompts you for a base directory and file name.

— **Download Mapping File**
Enables you to download the XML file from the Select Identity server if you are running the utility from a remote client. This displays a download dialog, which enables you to save the XML file locally.

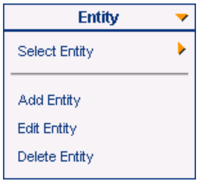
— **Download Reverse Synchronization File**
Enables you to download the XSL file from the Select Identity server if you are running the utility from a remote client. This displays a download dialog, which enables you to save the XSL file locally on the client.

  - **Reload**
    Reloads the XML file in the Attribute Mapping Utility.

  - **Disconnect**
    Disconnects the connection from the utility to the database.

- **Entity Menu**

  An entity is a logical grouping of attributes for users or groups (entitlements). By default, the User entity exists. If you wish to provision entitlements, you must create an additional entity. Each new entity is a group entity that represents user entitlements.



  Here are the options on this menu:

  - **Select Entity**
    Enables you to select the entity to edit.

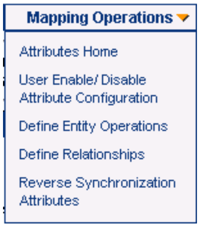
  - **Add Entity**
    Enables you to create a new entity in the utility, which enables you to map attributes for entities other than users. This displays a dialog prompting you to name the new entity.

  - **Edit Entity**
    Enables you to edit an entity name using this option. The default entity (user) cannot be edited.

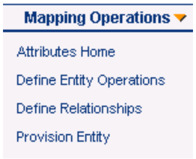
  - **Delete Entity**
    Deletes the currently selected entity. You cannot delete the default entity (user).

- **Mapping Operations Menu**

  This menu lists all of the operations that can be performed using the utility, which are the same operations through which you are guided if you use the wizard (see Mapping Pages on page 21). The following shows the menu when the user entity is selected:


Mapping Operations
Attributes Home
User Enable/ Disable Attribute Configuration
Define Entity Operations
Define Relationships
Reverse Synchronization Attributes

  Here is a snapshot of the menu when a group entity is selected:



  Here are the options on this menu:

  — **Attributes Home**
  Displays the Attributes page, which enables you to map database columns to Select Identity attributes.

  — **User Enable/Disable Attribute Configuration**
  Displays the Enable/Disable page, which enables you set values that are assigned when a user is enabled or disabled during provisioning.

  — **Define Entity Operations**
  Displays the Specify Supported Operations page, which enables you to define the operations that the connector can perform on the schema.

  — **Define Relationships**
  Displays the Define Relationship Definitions page, which enables you to define how tables in the schema relate.

  — **Reverse Synchronization Attributes**
  Displays Reverse Synchronization Attributes page, which enables you to map key fields that are used during reverse synchronization.

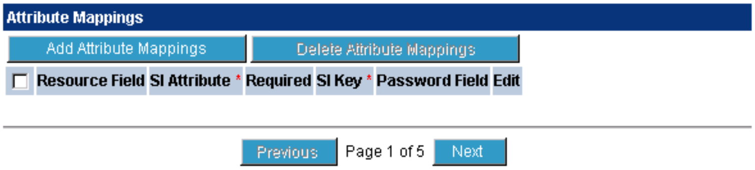
— **Provision Entity**
Displays the Provisioning Information page, which enable you to add or remove entitlements from the entitlement table.

- **Help Menu**

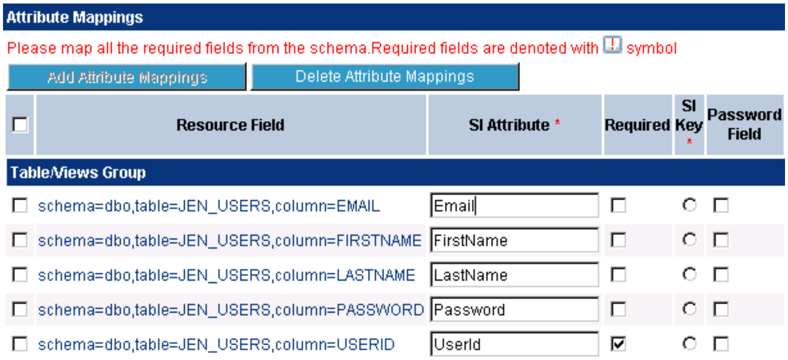
  This menu opens the help system, which provides an overview of the Attribute Mapping Utility.

Mapping Pages

By default, the Attributes page is displayed in the Attribute Mapping Utility window:



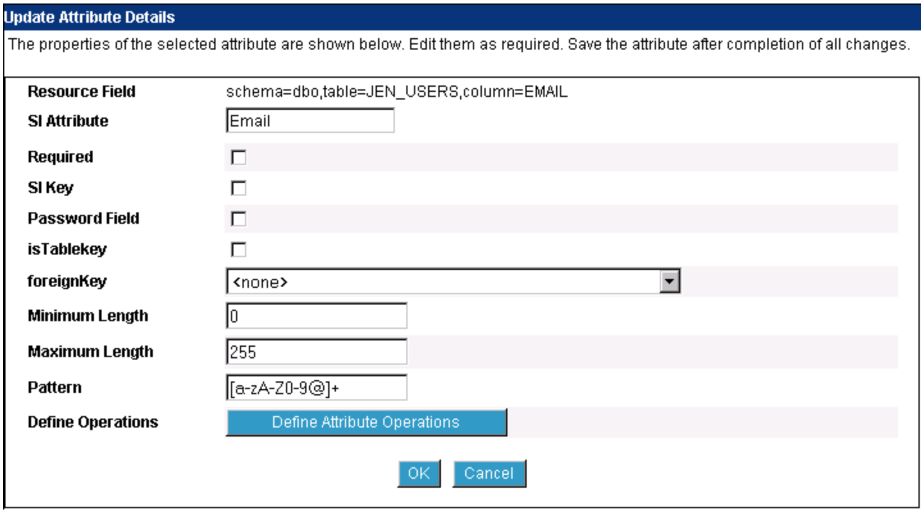
This page enables you to map database columns to Select Identity attributes. If you are creating a new mapping file, no mappings are listed. If you load an existing mapping file, mappings that are currently defined in the file are listed, as in this example:



Please map all the required fields from the schema.Required fields are denoted with ! symbol

☐	Resource Field	SI Attribute *	Required	SI Key *	Password Field
Table/Views Group					
☐	schema=dbo,table=JEN_USERS,column=EMAIL	Email	☐	○	☐
☐	schema=dbo,table=JEN_USERS,column=FIRSTNAME	FirstName	☐	○	☐
☐	schema=dbo,table=JEN_USERS,column=LASTNAME	LastName	☐	○	☐
☐	schema=dbo,table=JEN_USERS,column=PASSWORD	Password	☐	○	☐
☐	schema=dbo,table=JEN_USERS,column=USERID	UserId	☑	○	☐

Each database column listed under Resource Field is a link. If you click the link, the properties of that mapping are displayed:

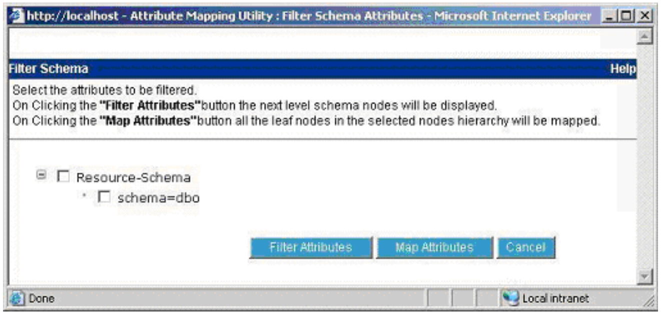


Update Attribute Details

The properties of the selected attribute are shown below. Edit them as required. Save the attribute after completion of all changes.

Resource Field	schema=dbo,table=JEN_USERS,column=EMAIL
SI Attribute	Email
Required	□
SI Key	□
Password Field	□
isTablekey	□
foreignKey	<none>
Minimum Length	0
Maximum Length	255
Pattern	[a-zA-Z0-9@]+
Define Operations	Define Attribute Operations

To access the database schema, click the **Add Attribute Mappings** button to display the Filter Schema page, which reads the database schema and provides a listing in a pop-up dialog. This displays the database schema in a hierarchical tree:



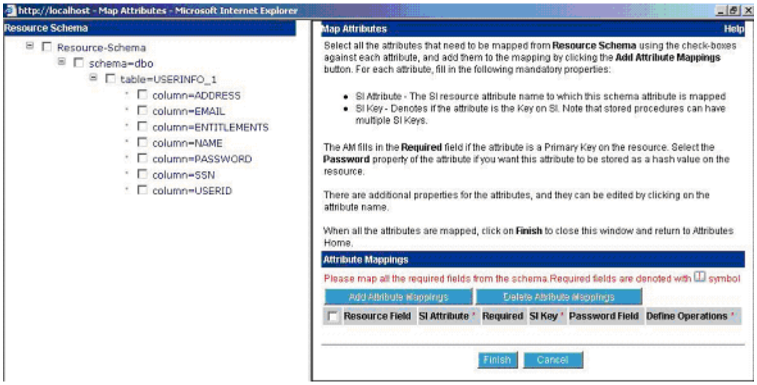
http://localhost - Attribute Mapping Utility : Filter Schema Attributes - Microsoft Internet Explorer

Filter Schema

Select the attributes to be filtered.
On Clicking the **"Filter Attributes"** button the next level schema nodes will be displayed.
On Clicking the **"Map Attributes"** button all the leaf nodes in the selected nodes hierarchy will be mapped.

□ Resource-Schema
□ schema=dbo

To map attributes, click the **Map Attributes** button to display the Map Attributes window, which reads the database schema and provides a listing on the left side of the window of the Selected Items from the Filter Schema page.



http://localhost - Map Attributes - Microsoft Internet Explorer

Resource Schema

- ⊟ □ Resource-Schema
  - ⊟ □ schema=dbo
    - ⊟ □ table=USERINFO_1
      - □ column=ADDRESS
      - □ column=EMAIL
      - □ column=ENTITLEMENTS
      - □ column=NAME
      - □ column=PASSWORD
      - □ column=SSN
      - □ column=USERID

Map Attributes — Help

Select all the attributes that need to be mapped from **Resource Schema** using the check-boxes against each attribute, and add them to the mapping by clicking the **Add Attribute Mappings** button. For each attribute, fill in the following mandatory properties:

- SI Attribute - The SI resource attribute name to which this schema attribute is mapped
- SI Key - Denotes if the attribute is the Key on SI. Note that stored procedures can have multiple SI Keys.

The AM fills in the **Required** field if the attribute is a Primary Key on the resource. Select the **Password** property of the attribute if you want this attribute to be stored as a hash value on the resource.

There are additional properties for the attributes, and they can be edited by clicking on the attribute name.

When all the attributes are mapped, click on **Finish** to close this window and return to Attributes Home.

Attribute Mappings

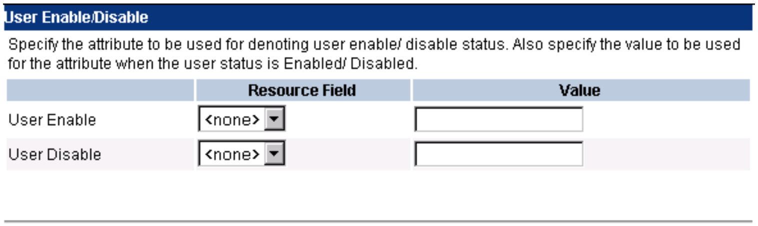
Please map all the required fields from the schema.Required fields are denoted with ! symbol

□	Resource Field	SI Attribute *	Required	SI Key *	Password Field	Define Operations *

Note that if you are simply editing a mapping file and you are not connected to the database, the schema is not displayed.

Click the **Previous** and **Next** buttons at the bottom of the Attribute Mapping Utility window or select options on the Mapping Operations menu to display the following pages of the wizard:

— **User Enable/Disable** page:
Enables you to set values that are assigned when a user is enabled or disabled during provisioning.

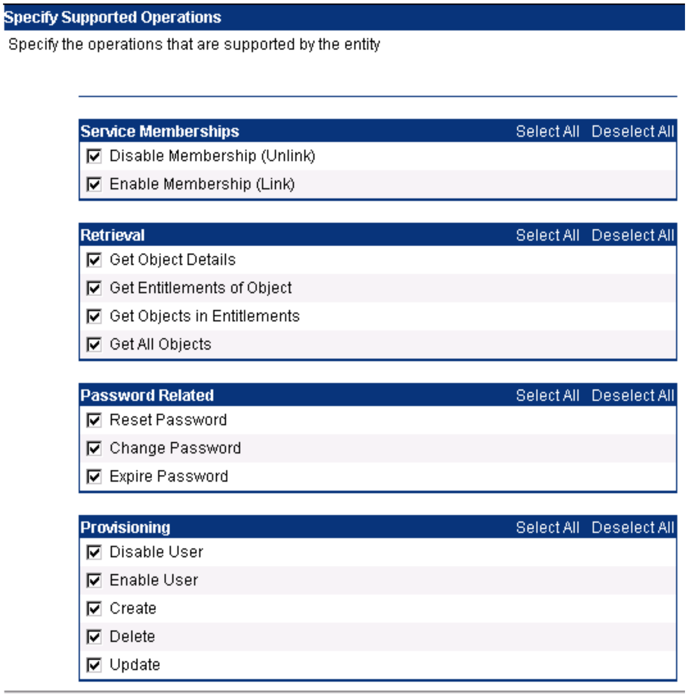


User Enable/Disable

Specify the attribute to be used for denoting user enable/ disable status. Also specify the value to be used for the attribute when the user status is Enabled/ Disabled.

	Resource Field	Value
User Enable	<none>	
User Disable	<none>	

— **Specify Supported Operations** page:
Enables you to define the operations that the connector can perform on the schema.



Specify Supported Operations

Specify the operations that are supported by the entity

Service Memberships Select All Deselect All

☑ Disable Membership (Unlink)
☑ Enable Membership (Link)

Retrieval Select All Deselect All

☑ Get Object Details
☑ Get Entitlements of Object
☑ Get Objects in Entitlements
☑ Get All Objects

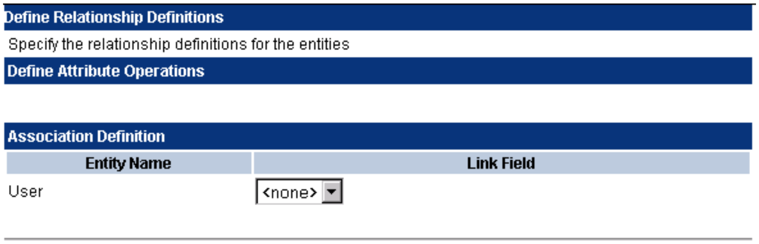
Password Related Select All Deselect All

☑ Reset Password
☑ Change Password
☑ Expire Password

Provisioning Select All Deselect All

☑ Disable User
☑ Enable User
☑ Create
☑ Delete
☑ Update

— **Define Relationship Definitions** page:
Enables you to define how tables in the schema relate.



Define Relationship Definitions

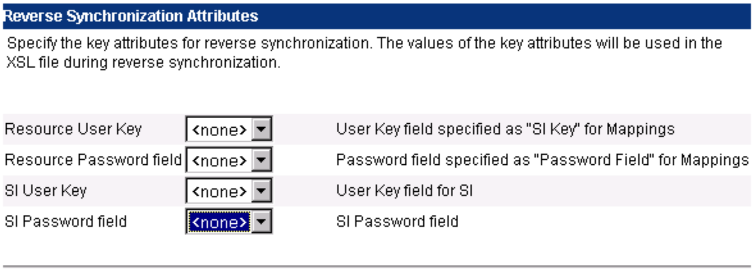
Specify the relationship definitions for the entities

Define Attribute Operations

Association Definition

Entity Name	Link Field
User	<none> ▾

— **Reverse Synchronization Attributes** page:
Enables you to map key fields that are used during reverse synchronization.

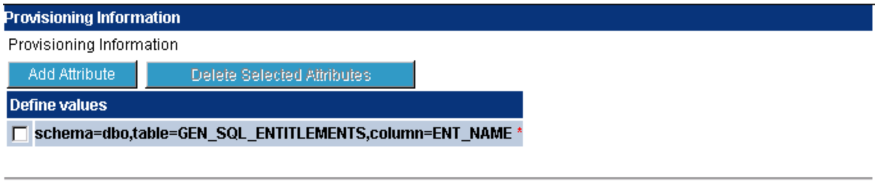


Reverse Synchronization Attributes

Specify the key attributes for reverse synchronization. The values of the key attributes will be used in the XSL file during reverse synchronization.

Resource User Key	<none>	User Key field specified as "SI Key" for Mappings
Resource Password field	<none>	Password field specified as "Password Field" for Mappings
SI User Key	<none>	User Key field for SI
SI Password field	<none>	SI Password field

— **Provisioning Information** page:
Displays the attributes that are mapped for the group entity, except for the attribute that is linked to the user attribute. You can provision entitlements directly in to the database. Note that the Attribute Mapping Utility provisions group entities only.



Provisioning Information

Provisioning Information

Define values

□ schema=dbo,table=GEN_SQL_ENTITLEMENTS,column=ENT_NAME *

See Attribute Mappings on page 26 for procedures that illustrate how to set properties in these sections of the window.

2

Attribute Mappings

This chapter describes how to use the Attribute Mapping Utility to map database columns to Select Identity attributes. You can also define relationships between columns and specify the operations that can be performed by the connector.

After you define the mappings, the resulting XML and XSL files are stored in the location specified when you connected to the database (when you logged in to the utility).

Usage Scenario in This Chapter

The database schema that stores user and entitlement data can be created in a number of ways; Usage Scenarios on page 63 describes the most common schema configurations. The scenarios used in this chapter are the more likely configurations and provide a broader view of the required steps to define mappings for the XML and XSL files.

Table Scenario

For the user and entitlement provisioning procedures, three tables are used to store provisioning data:

- A user table called GEN_SQL_USERS that contains six columns, which define user attributes. For example, the table may contain the UserName, Password, Firstname, Lastname, Email, and City columns, which store user-related information.
- An entitlement table called GEN_SQL_ENTITLEMENTS that contains one column, which stores the list of possible entitlements that can be assigned to users. For example, the table may contain the EntitlementID and Description columns, which store entitlement information that can be assigned to users.
- A normalized mapping table called GEN_SQL_USER_TO_ENT that defines user-to-entitlement mappings and contains two columns, one to store user IDs and one to store the assigned entitlements. For example, the table may contain the UserName and EntitlementID columns, which store user name-to-entitlement mappings.

The user column in the mapping table uses the user column in the user table as its foreign key. Similarly, the entitlements column in the mapping table uses the entitlement column in the user table as its foreign key. Finally, when users and entitlements are provisioned, there can be one user ID per user and zero or more entitlements per user.

Stored Procedure Scenario

The database connectors allow the usage of stored procedures for provisioning and other user-related operations. A set of stored procedures can be written to perform operations that are typically done by the connector. These operations include creating a user, modifying a user, deleting a user, and so on. Hence, a stored procedure provides a way to abstract connector operations.

Stored procedures can be customized for the necessary operations; Usage Scenarios on page 63 describes the most common schema configurations. The scenario used in this chapter is one of the more likely configurations. Here is a description of the scenario:

Three stored procedures are created in the database:

- The addUser stored procedure can create a user in the database, in the Users table. Values are passed as parameters to the procedure.
- The modifyUser stored procedure can modify user attributes for an existing user in the database. This procedure modifies all columns in the Users table except the UserName and Password columns.
- The deleteUser stored procedure takes a user name as a parameter and deletes the user from the database table.

Defining User Mappings

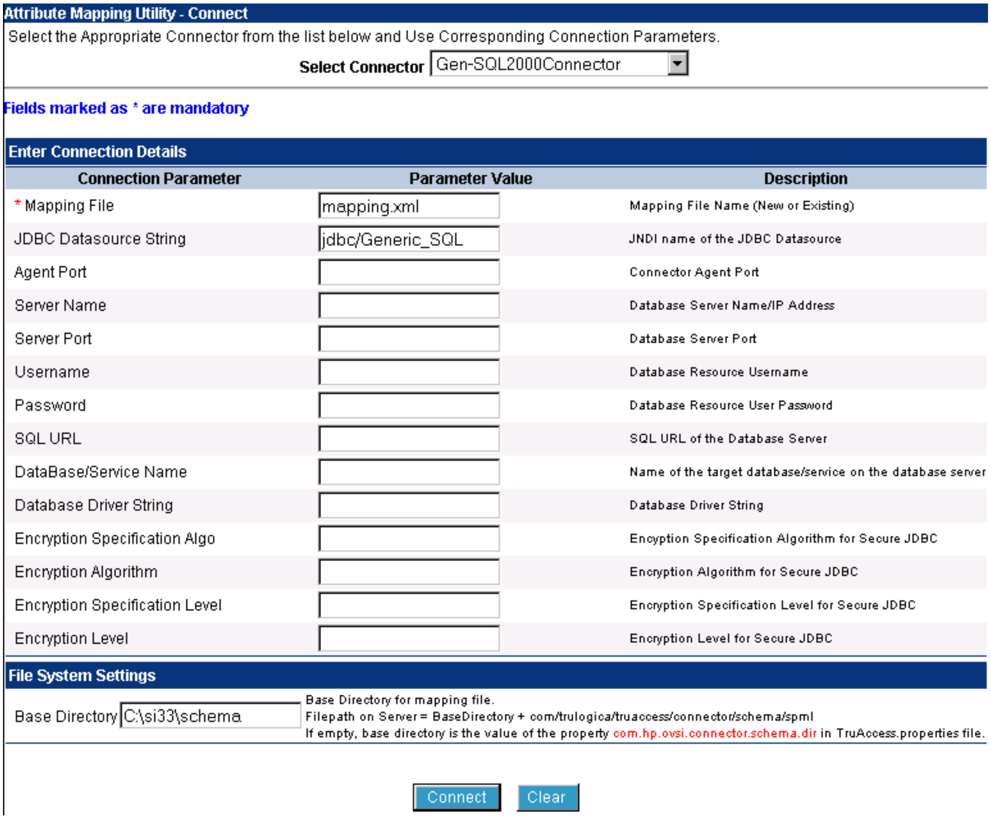
The following procedure describes how to map user and entitlement attributes to Select Identity attributes. When you finish defining the mappings, the XML mapping file is generated in the directory specified in the Base Directory field when you logged in to the Attribute Mapping Utility and connected to the schema.

1. Access the Attribute Mapping Utility by entering the following URL in the browser:

   http://*hostname*:*port*/attributemapper/index.jsp

   where *hostname* is the name or IP address of the application server on which Select Identity is deployed and *port* is the application server's port.

2 Connect to the database by selecting the deployed connector from the Select Connector drop-down list and entering connection information for the database. For this scenario, a JDBC data source is used to connect to the database, therefore only the mapping file name and JDBC data source are specified. The Base Directory field specifies where the mapping files are stored on the Select Identity server.



Attribute Mapping Utility - Connect

Select the Appropriate Connector from the list below and Use Corresponding Connection Parameters.

Select Connector: Gen-SQL2000Connector

Fields marked as * are mandatory

Enter Connection Details

Connection Parameter	Parameter Value	Description
* Mapping File	mapping.xml	Mapping File Name (New or Existing)
JDBC Datasource String	jdbc/Generic_SQL	JNDI name of the JDBC Datasource
Agent Port		Connector Agent Port
Server Name		Database Server Name/IP Address
Server Port		Database Server Port
Username		Database Resource Username
Password		Database Resource User Password
SQL URL		SQL URL of the Database Server
DataBase/Service Name		Name of the target database/service on the database server
Database Driver String		Database Driver String
Encryption Specification Algo		Encyption Specification Algorithm for Secure JDBC
Encryption Algorithm		Encryption Algorithm for Secure JDBC
Encryption Specification Level		Encryption Specification Level for Secure JDBC
Encryption Level		Encryption Level for Secure JDBC

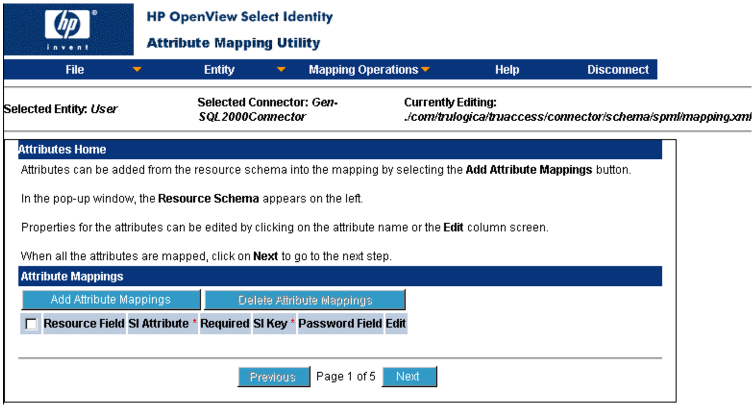
File System Settings

Base Directory: C:\si33\schema

Base Directory for mapping file.
Filepath on Server = BaseDirectory + com/trulogica/truaccess/connector/schema/spml
If empty, base directory is the value of the property com.hp.ovsi.connector.schema.dir in TruAccess.properties file.

Then, click **Connect**.

If the utility successfully connects to the database, the Attributes page displays, which enables you to map user attributes. By default, the user entity is selected.


HP OpenView Select Identity
Attribute Mapping Utility
File
Entity
Mapping Operations
Help
Disconnect
Add Attribute Mappings
Delete Attribute Mappings
Previous
Next

Selected Entity: *User*

Selected Connector: *Gen-SQL2000Connector*

Currently Editing: *./com/trulogica/truaccess/connector/schema/spml/mapping.xml*

Attributes Home

Attributes can be added from the resource schema into the mapping by selecting the **Add Attribute Mappings** button.

In the pop-up window, the **Resource Schema** appears on the left.

Properties for the attributes can be edited by clicking on the attribute name or the **Edit** column screen.

When all the attributes are mapped, click on **Next** to go to the next step.

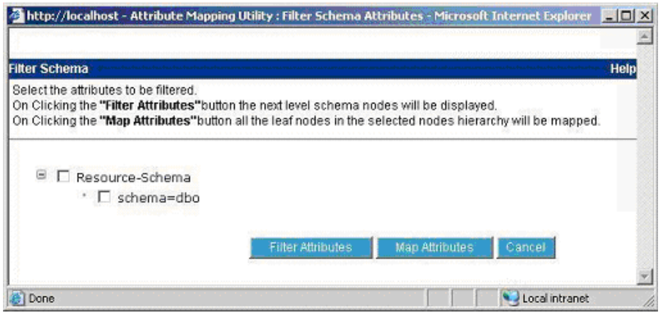
Attribute Mappings

□	Resource Field	SI Attribute *	Required	SI Key *	Password Field	Edit

Page 1 of 5

3 Map each user attribute defined in the database schema to a Select Identity attribute:

a Click **Add Attribute Mappings** to display the Filter Schema page, which provides a listing of the database schema in a pop-up dialog. This displays the database schema in a hierarchical form.


http://localhost - Attribute Mapping Utility : Filter Schema Attributes - Microsoft Internet Explorer
Help
Filter Attributes
Map Attributes
Cancel
Done
Local intranet

Filter Schema

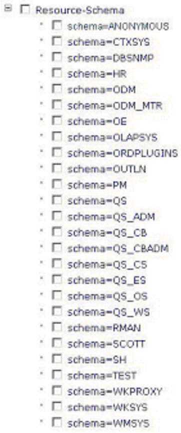
Select the attributes to be filtered.
On Clicking the **"Filter Attributes"**button the next level schema nodes will be displayed.
On Clicking the **"Map Attributes"**button all the leaf nodes in the selected nodes hierarchy will be mapped.

□ Resource-Schema
□ schema=dbo

From this page, you can filter the tables, views, and stored procedures that you wish to map from the the rest of the database schema. This reduces the schema retrieval time and also provides a better view by showing only the selected schema when mapping attributes.

Note the following points when using Attribute Mapping Utility for Oracle:

– The Filter Schema page shows the schema for all users in Oracle. Here is an example of the Oracle schema displayed on the Filter Schema page:

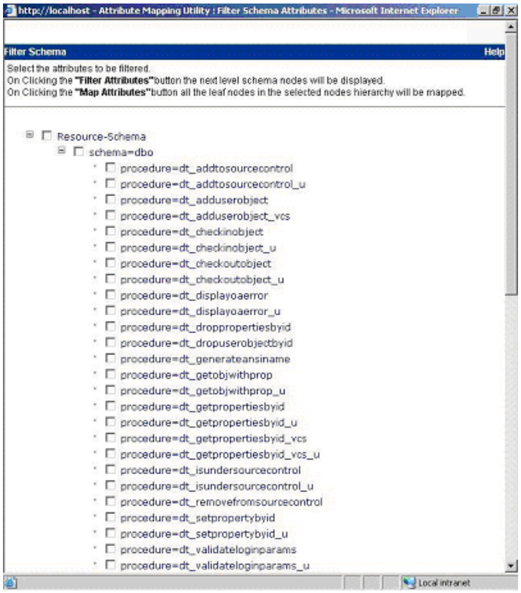


⊟ □ Resource-Schema
- □ schema=ANONYMOUS
- □ schema=CTXSYS
- □ schema=DBSNMP
- □ schema=HR
- □ schema=ODM
- □ schema=ODM_MTR
- □ schema=OE
- □ schema=OLAPSYS
- □ schema=ORDPLUGINS
- □ schema=OUTLN
- □ schema=PM
- □ schema=QS
- □ schema=QS_ADM
- □ schema=QS_CB
- □ schema=QS_CBADM
- □ schema=QS_CS
- □ schema=QS_ES
- □ schema=QS_OS
- □ schema=QS_WS
- □ schema=RMAN
- □ schema=SCOTT
- □ schema=SH
- □ schema=TEST
- □ schema=WKPROXY
- □ schema=WKSYS
- □ schema=WMSYS

– Select the schema where the required tables are present and click **Filter Attribute** to show only the tables or stored procedures in that schema.

– Make sure that the you have access privileges for the selected schema on the Filter Schema' page. If not, the Map Attributes window will not display a schema. If more than one schema is selected and access privileges are available for a subset of the selected schemas, only the subset of schemas is displayed on the Map Attributes window.

b To select part of the schema for mapping, select the **Schema** check box and click the **Filter Attributes** button to display the schema. (You can also simply click the **Map Attributes** button to view the entire schema when mapping attributes.)



http://localhost - Attribute Mapping Utility : Filter Schema Attributes - Microsoft Internet Explorer

Filter Schema — Help

Select the attributes to be filtered.
On Clicking the **"Filter Attributes"**button the next level schema nodes will be displayed.
On Clicking the **"Map Attributes"**button all the leaf nodes in the selected nodes hierarchy will be mapped.

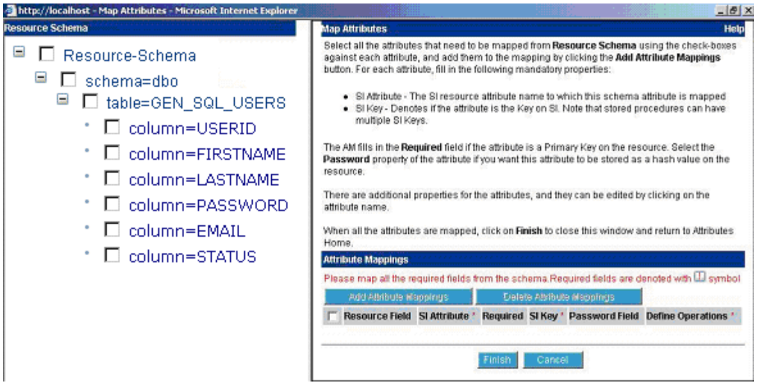
```
⊟ □ Resource-Schema
    ⊟ □ schema=dbo
        · □ procedure=dt_addtosourcecontrol
        · □ procedure=dt_addtosourcecontrol_u
        · □ procedure=dt_adduserobject
        · □ procedure=dt_adduserobject_vcs
        · □ procedure=dt_checkinobject
        · □ procedure=dt_checkinobject_u
        · □ procedure=dt_checkoutobject
        · □ procedure=dt_checkoutobject_u
        · □ procedure=dt_displayoaerror
        · □ procedure=dt_displayoaerror_u
        · □ procedure=dt_droppropertiesbyid
        · □ procedure=dt_dropuserobjectbyid
        · □ procedure=dt_generateansiname
        · □ procedure=dt_getobjwithprop
        · □ procedure=dt_getobjwithprop_u
        · □ procedure=dt_getpropertiesbyid
        · □ procedure=dt_getpropertiesbyid_u
        · □ procedure=dt_getpropertiesbyid_vcs
        · □ procedure=dt_getpropertiesbyid_vcs_u
        · □ procedure=dt_isundersourcecontrol
        · □ procedure=dt_isundersourcecontrol_u
        · □ procedure=dt_removefromsourcecontrol
        · □ procedure=dt_setpropertybyid
        · □ procedure=dt_setpropertybyid_u
        · □ procedure=dt_validateloginparams
        · □ procedure=dt_validateloginparams_u
```

The database schema is shown on the Filter Schema page in a tree that you can expand.

Here is an explanation of the tree:

```
Database Name
    |
    | - Schema Names
    |      | - Table Names
    |      | - Procedure Names
    |      |  - View Names
```

c Select the user tables that you wish to map. After you select tables, click on the **Map Attributes** button. This displays the Map Attributes window, which displays the expanded schema items selected on Filter Schema page.



http://localhost - Map Attributes - Microsoft Internet Explorer

Resource Schema

- □ Resource-Schema
  - □ schema=dbo
    - □ table=GEN_SQL_USERS
      - □ column=USERID
      - □ column=FIRSTNAME
      - □ column=LASTNAME
      - □ column=PASSWORD
      - □ column=EMAIL
      - □ column=STATUS

Map Attributes | Help

Select all the attributes that need to be mapped from **Resource Schema** using the check-boxes against each attribute, and add them to the mapping by clicking the **Add Attribute Mappings** button. For each attribute, fill in the following mandatory properties:

- SI Attribute - The SI resource attribute name to which this schema attribute is mapped
- SI Key - Denotes if the attribute is the Key on SI. Note that stored procedures can have multiple SI Keys.

The AM fills in the **Required** field if the attribute is a Primary Key on the resource. Select the **Password** property of the attribute if you want this attribute to be stored as a hash value on the resource.

There are additional properties for the attributes, and they can be edited by clicking on the attribute name.

When all the attributes are mapped, click on **Finish** to close this window and return to Attributes Home.

Attribute Mappings

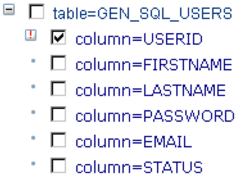
Please map all the required fields from the schema.Required fields are denoted with ! symbol

□	Resource Field	SI Attribute *	Required	SI Key *	Password Field	Define Operations *

Here is the explanation of the tree:

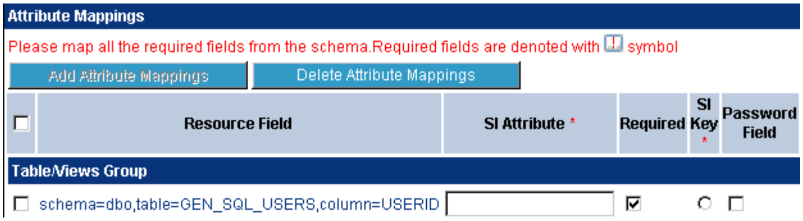
```
Database Name
   |
   | - Schema Names
   |     | - Table Names
   |     |     | - Column Names
   |     | - Procedure Names
   |     |     | - In Parameters
   |     |     | - Out Parameters
```

d From the left side of the window, select the database column that you would like to map to a Select Identity attribute:



- □ table=GEN_SQL_USERS
  - ! ☑ column=USERID
  - □ column=FIRSTNAME
  - □ column=LASTNAME
  - □ column=PASSWORD
  - □ column=EMAIL
  - □ column=STATUS

Then, click the **Add Attribute Mappings** button. The selected column is added to the table:



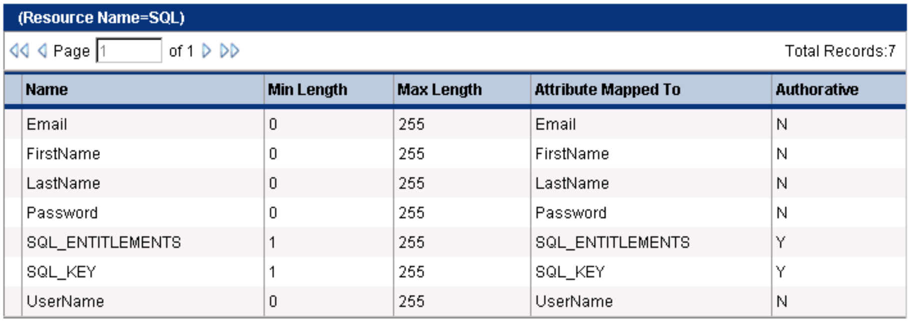
Attribute Mappings

Please map all the required fields from the schema.Required fields are denoted with ! symbol

□	Resource Field	SI Attribute *	Required	SI Key *	Password Field
Table/Views Group					
□	schema=dbo,table=GEN_SQL_USERS,column=USERID		☑	○	□

If you map a stored procedure, another section named Stored Procedure Group is added to the page below the Table/Views Group section.

e In the SI Attribute field, enter the name of the Select Identity attribute to which you want to map the selected database column. This value is the same as the value displayed in the Name column when creating attributes for the connector’s resource on the Select Identity client. Here is a snapshot of the resource attributes for this scenario:

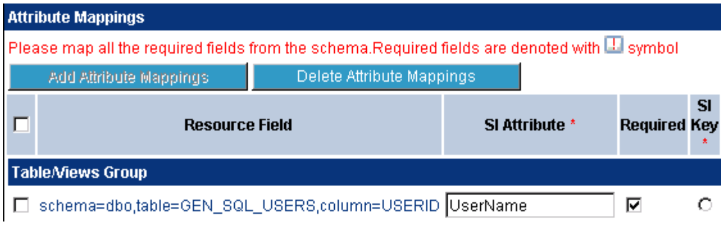


(Resource Name=SQL)

Page 1 of 1 Total Records:7

Name	Min Length	Max Length	Attribute Mapped To	Authorative
Email	0	255	Email	N
FirstName	0	255	FirstName	N
LastName	0	255	LastName	N
Password	0	255	Password	N
SQL_ENTITLEMENTS	1	255	SQL_ENTITLEMENTS	Y
SQL_KEY	1	255	SQL_KEY	Y
UserName	0	255	UserName	N

Because UserName is the attribute name in this scenario, enter **`UserName`** in the SI Attribute field for the USERID column:



Attribute Mappings

Please map all the required fields from the schema.Required fields are denoted with ! symbol

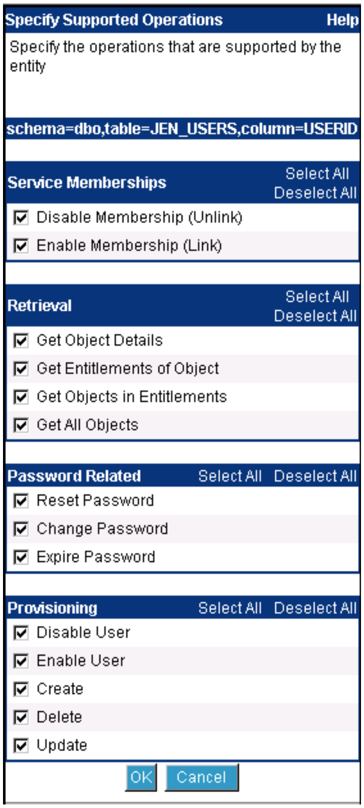
☐	Resource Field	SI Attribute *	Required	SI Key *
Table/Views Group				
☐	schema=dbo,table=GEN_SQL_USERS,column=USERID	UserName	☑	○

The **Required** option is selected if the chosen database column is the schema's primary key.

f Select the **SI Key** option if the specified attribute is a key field in Select Identity. If you are adding a stored procedure, you can select multiple attributes as key fields.

g If you wish to encrypt the value of the attribute in the database, such as if an attribute can be used in the Reset Password or Change Password operation in Select Identity, select the **Password Field** option.

h If you wish to define the actions that the connector can perform for the attribute (not the entire entity), click the **Define Attribute Operations** button and select the allowed operations. A pop-up window displays in which you can select the operations then click **OK**.



Specify Supported Operations Help

Specify the operations that are supported by the entity

schema=dbo,table=JEN_USERS,column=USERID

Service Memberships Select All Deselect All

☑ Disable Membership (Unlink)
☑ Enable Membership (Link)

Retrieval Select All Deselect All

☑ Get Object Details
☑ Get Entitlements of Object
☑ Get Objects in Entitlements
☑ Get All Objects

Password Related Select All Deselect All

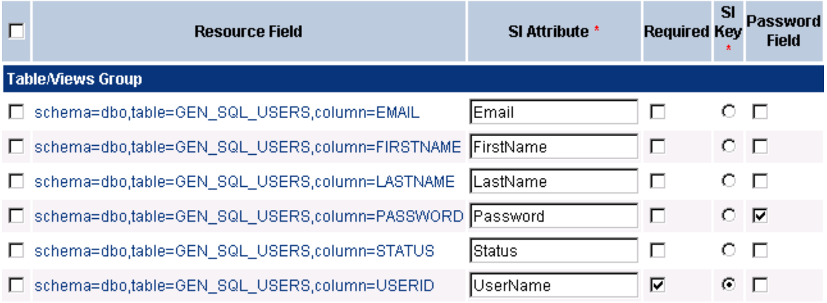
☑ Reset Password
☑ Change Password
☑ Expire Password

Provisioning Select All Deselect All

☑ Disable User
☑ Enable User
☑ Create
☑ Delete
☑ Update

OK Cancel

i Repeat Step d through Step h for each user column you wish to map. The columns are added to the right side of the page in alphabetical order. Here is a snapshot of the defined user mappings for this scenario:

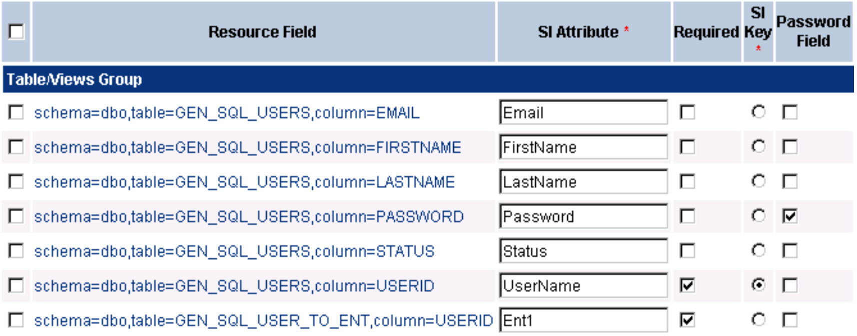


□	Resource Field	SI Attribute *	Required	SI Key *	Password Field
Table/Views Group					
□	schema=dbo,table=GEN_SQL_USERS,column=EMAIL	Email	□	○	□
□	schema=dbo,table=GEN_SQL_USERS,column=FIRSTNAME	FirstName	□	○	□
□	schema=dbo,table=GEN_SQL_USERS,column=LASTNAME	LastName	□	○	□
□	schema=dbo,table=GEN_SQL_USERS,column=PASSWORD	Password	□	○	☑
□	schema=dbo,table=GEN_SQL_USERS,column=STATUS	Status	□	○	□
□	schema=dbo,table=GEN_SQL_USERS,column=USERID	UserName	☑	●	□

Note that the Select Identity attribute specified for the STATUS column is arbitrary; you must map the column but the Enable/Disable settings (Step 4) control what is provisioned to this column.

j If you will provision entitlements and the database schema includes two tables (one for user data and one for entitlements), you must map the user column from the entitlement table. If the database schema includes at least three tables, including a user-to-entitlement mapping table, you must map the columns of this table accordingly.

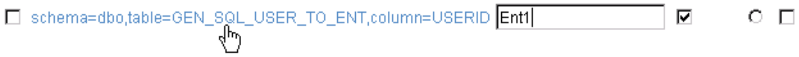
Thus, as shown in the following snapshot, map the USERID column from the GEN_SQL_USER_TO_ENT table for the User entity:



□	Resource Field	SI Attribute *	Required	SI Key *	Password Field
Table/Views Group					
□	schema=dbo,table=GEN_SQL_USERS,column=EMAIL	Email	□	○	□
□	schema=dbo,table=GEN_SQL_USERS,column=FIRSTNAME	FirstName	□	○	□
□	schema=dbo,table=GEN_SQL_USERS,column=LASTNAME	LastName	□	○	□
□	schema=dbo,table=GEN_SQL_USERS,column=PASSWORD	Password	□	○	☑
□	schema=dbo,table=GEN_SQL_USERS,column=STATUS	Status	□	○	□
□	schema=dbo,table=GEN_SQL_USERS,column=USERID	UserName	☑	●	□
□	schema=dbo,table=GEN_SQL_USER_TO_ENT,column=USERID	Ent1	☑	○	□

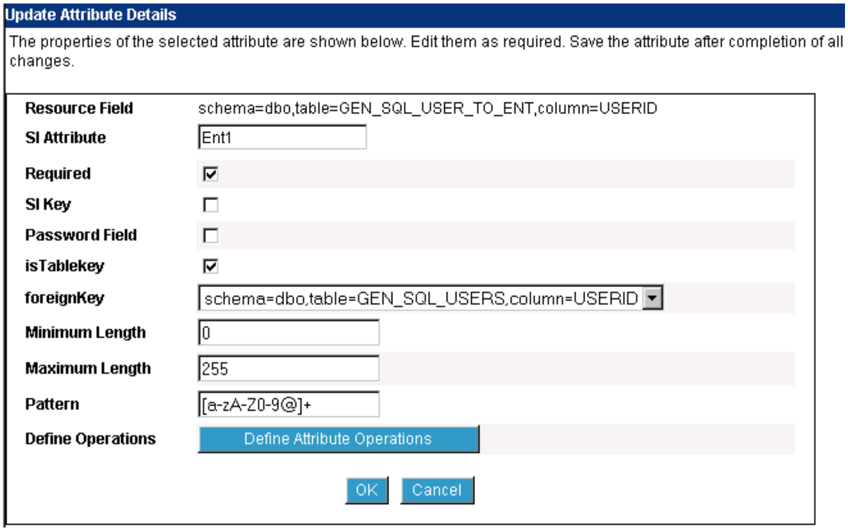
Note that you can specify any value in the SI Attribute field for this mapping.

The user column in the mapping table is a foreign key to the USERID column of the GEN_SQL_USERS table. You must specify this for the mapping. Click the column name in the Resource Field:



□ schema=dbo,table=GEN_SQL_USER_TO_ENT,column=USERID Ent1 ☑ ○ □

The Update Attribute Details page displays. From the foreignKey drop-down list, select the user column of the user table. Also, select the **isTableKey** option.



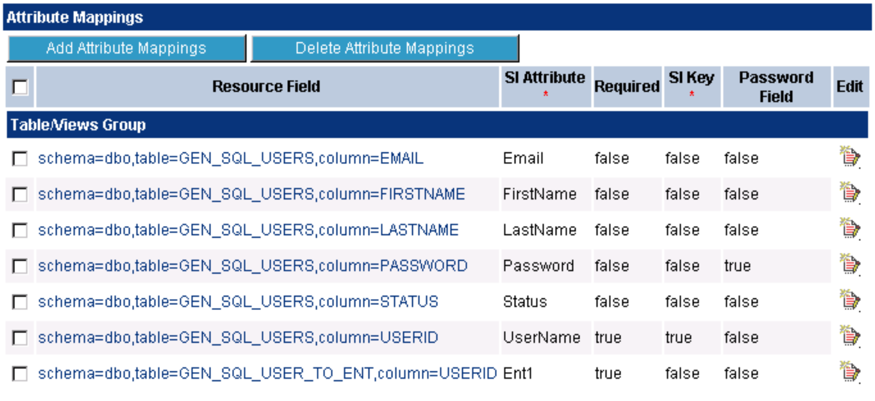
Update Attribute Details

The properties of the selected attribute are shown below. Edit them as required. Save the attribute after completion of all changes.

Resource Field	schema=dbo,table=GEN_SQL_USER_TO_ENT,column=USERID
SI Attribute	Ent1
Required	☑
SI Key	□
Password Field	□
isTablekey	☑
foreignKey	schema=dbo,table=GEN_SQL_USERS,column=USERID ▾
Minimum Length	0
Maximum Length	255
Pattern	[a-zA-Z0-9@]+
Define Operations	Define Attribute Operations

Then, click **OK**.

k Click **Finish** at the bottom of the Map Attributes window. The window closes and the mapped attributes are listed on the Attribute Mapping Utility page:

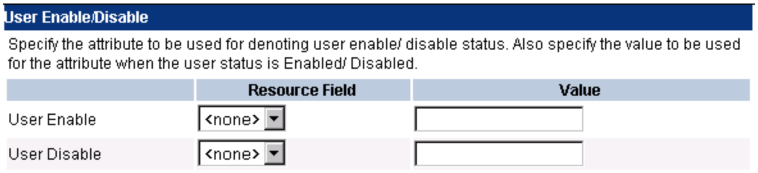


Attribute Mappings

□	Resource Field	SI Attribute *	Required	SI Key *	Password Field	Edit
	Table/Views Group					
□	schema=dbo,table=GEN_SQL_USERS,column=EMAIL	Email	false	false	false	✎
□	schema=dbo,table=GEN_SQL_USERS,column=FIRSTNAME	FirstName	false	false	false	✎
□	schema=dbo,table=GEN_SQL_USERS,column=LASTNAME	LastName	false	false	false	✎
□	schema=dbo,table=GEN_SQL_USERS,column=PASSWORD	Password	false	false	true	✎
□	schema=dbo,table=GEN_SQL_USERS,column=STATUS	Status	false	false	false	✎
□	schema=dbo,table=GEN_SQL_USERS,column=USERID	UserName	true	true	false	✎
□	schema=dbo,table=GEN_SQL_USER_TO_ENT,column=USERID	Ent1	true	false	false	✎

4 Define the Enable and Disable settings for the connector. This enables you to define the attribute that will be used for denoting user status. Complete the following steps:

a Click **Next** under the Attribute Mappings section of the page or select **Mapping Operations →User Enable/Disable Attribute Configuration**. The User Enable/Disable section is displayed:



User Enable/Disable

Specify the attribute to be used for denoting user enable/ disable status. Also specify the value to be used for the attribute when the user status is Enabled/ Disabled.

	Resource Field	Value
User Enable	<none> ▾	
User Disable	<none> ▾	

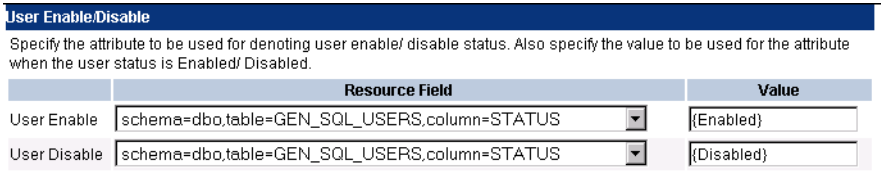
b In the User Enable row, select a database table from the Resource Field drop-down list that will store a status value if the user is enabled. This column will be used by Select Identity's Enable All operation.

c Specify the value to be stored when the user is enabled. If you intend to provision static text, surround the specified value with brackets ({ }).

d In the User Disable row, select a database table from the Resource Field drop-down list that will store a status value if the user is disabled. This column will be used by Select Identity's Disable All operation.

e Specify the value to be stored when the user is disabled. If you intend to provision static text, surround the specified value with brackets ({ }).

Here is a snapshot of the values specified for this scenario:



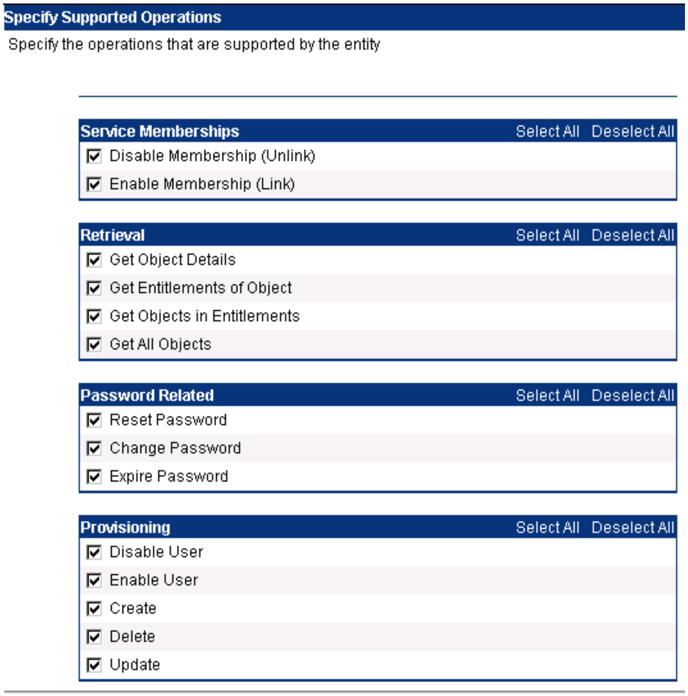
User Enable/Disable

Specify the attribute to be used for denoting user enable/ disable status. Also specify the value to be used for the attribute when the user status is Enabled/ Disabled.

	Resource Field	Value
User Enable	schema=dbo,table=GEN_SQL_USERS,column=STATUS	{Enabled}
User Disable	schema=dbo,table=GEN_SQL_USERS,column=STATUS	{Disabled}

5 Define the operations that Select Identity can perform on the database schema:

a Click **Next** under the User Enable/Disable section of the page or select **Mapping Operations →Define Entity Operations**. The Specify Supported Operations section is displayed:



Specify Supported Operations

Specify the operations that are supported by the entity

Service Memberships

☑ Disable Membership (Unlink)

☑ Enable Membership (Link)

Retrieval

☑ Get Object Details

☑ Get Entitlements of Object

☑ Get Objects in Entitlements

☑ Get All Objects

Password Related

☑ Reset Password

☑ Change Password

☑ Expire Password

Provisioning

☑ Disable User

☑ Enable User

☑ Create

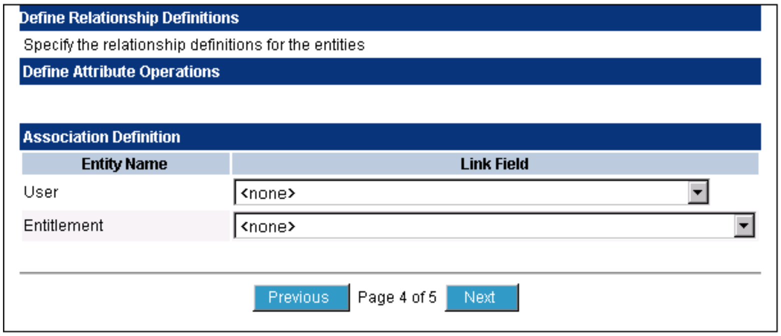
☑ Delete

☑ Update

b Select or deselect the operations you want the connector to perform when provisioning users in the database schema (in the user table).

6 Define the relationship between entities and database tables by completing the following steps:

a Click **Next** under the Specify Supported Operations section of the page or select **Mapping Operations** →**Define Relationships**. The Define Relationship Definitions section is displayed:



Define Relationship Definitions

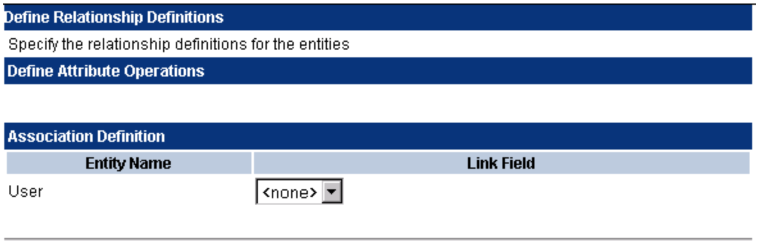
Specify the relationship definitions for the entities

Define Attribute Operations

Association Definition

Entity Name	Link Field
User	<none>
Entitlement	<none>

If the entitlement entity has not been created in the Attribute Mapping Utility, the entitlement entity is not displayed here:



Define Relationship Definitions

Specify the relationship definitions for the entities

Define Attribute Operations

Association Definition

Entity Name	Link Field
User	<none>

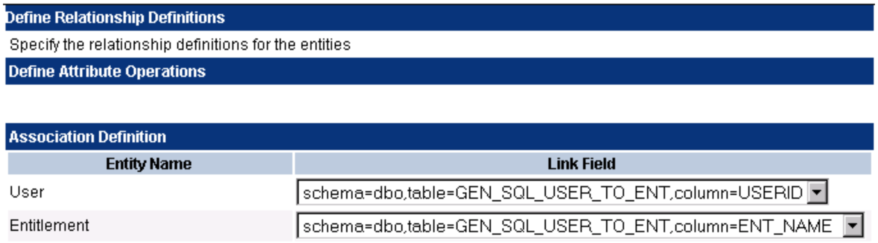
If you wish to provision entitlements, complete the steps in Defining Entitlement Mappings on page 45 then return to this step.

b For the user entity, select the column that will store the User entity's Key field for operations that involve entitlements. In this scenario, the user column in the mapping table is selected.

If there is no entitlement table and the connector will not perform group or entitlement operations, you can leave this field empty. If a column is selected from the Link Field drop-down list, it will not be available for provisioning user information.

c For the entitlement entity, select the column that will store the user's entitlements. In this scenario, select the entitlement column in the mapping table.

Here is a snapshot of the relationships for the example scenario:



Define Relationship Definitions

Specify the relationship definitions for the entities

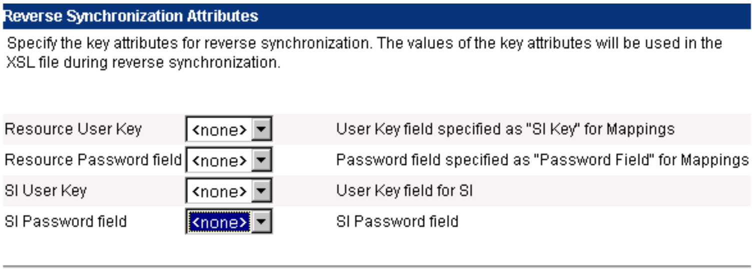
Define Attribute Operations

Association Definition

Entity Name	Link Field
User	schema=dbo,table=GEN_SQL_USER_TO_ENT,column=USERID
Entitlement	schema=dbo,table=GEN_SQL_USER_TO_ENT,column=ENT_NAME

7 If you have configured the agent to perform reverse synchronization, you must define reverse mappings:

a Click **Next** under the Define Relationship Definitions section of the page or select **Mapping Operations** →**Reverse Synchronization Attributes**. The Reverse Synchronization Attributes section is displayed:



Reverse Synchronization Attributes

Specify the key attributes for reverse synchronization. The values of the key attributes will be used in the XSL file during reverse synchronization.

Resource User Key	<none>	User Key field specified as "SI Key" for Mappings
Resource Password field	<none>	Password field specified as "Password Field" for Mappings
SI User Key	<none>	User Key field for SI
SI Password field	<none>	SI Password field

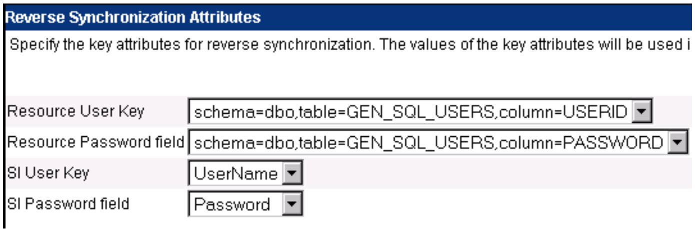
b From the Resource User Key drop-down list, select the database column that was mapped to the Select Identity attribute that is the Select Identity key.

c From the Resource Password field drop-down list, select the database column that was mapped to the Select Identity password attribute.

d From the SI User Key drop-down list, select the Select Identity attribute that is the Select Identity key for the user.

e From the SI Password field drop-down list, select the Select Identity attribute that stores the user password.

Here is a snapshot of the values specified for this scenario:



Reverse Synchronization Attributes

Specify the key attributes for reverse synchronization. The values of the key attributes will be used i

Resource User Key	schema=dbo,table=GEN_SQL_USERS,column=USERID
Resource Password field	schema=dbo,table=GEN_SQL_USERS,column=PASSWORD
SI User Key	UserName
SI Password field	Password

8 Select **File →Save Mapping File** to generate the XML and XSL files in the location specified in the Base Directory field when you connected.

You may want to verify that the `com.hp.ovsi.connector.schema.dir` property is set in the `TruAccess.properties` file, which resides in the *`install_dir`*`/sysArchive` directory on the Select Identity server. It should be set to the top-level directory where the mapping files will be created, which is specified in the Base Directory field on the Attribute Mapping Utility interface.

If no value was specified in the Base Directory field, the `com/trulogica/truaccess/connector/schema/spml` directory structure was created in the application server's home directory.

For example, if you entered `C:\SI_3.3\schema` in the Base Directory field, the `com/trulogica/truaccess/connector/schema/spml` directory structure was created under that directory, and the XML and XSL files were created there. In this example, the files reside in this directory:

```
C:\SI3.3\schema\com\trulogica\truaccess\connector\schema\
spml
```

Thus, you would set the `com.hp.ovsi.connector.schema.dir` property as follows:

```
com.hp.ovsi.connector.schema.dir = C:/SI3.3/schema
```

If you wish to download the XML and XSL files to the local system, select **File →Download Mapping File** or **File →Download Reverse Synchronization File**, respectively. To disconnect from the database or log out of the utility, click **Disconnect**.

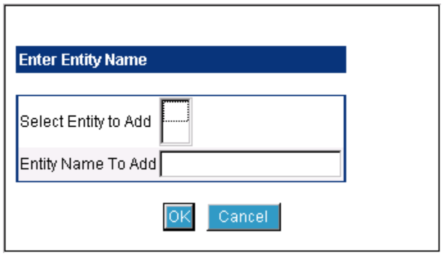
Defining Entitlement Mappings

If the database schema contains entitlement data and you wish to provision entitlements, you must create an entitlement entity in the Attribute Mapping Utility. Otherwise, performing the steps in Defining User Mappings on page 28 generates the XML and XSL files required to provision users only.

Complete the following steps to create an entitlement entity and map its attributes. This procedure assumes that you are logged in to the Attribute Mapping Utility.

1 Add the entitlement entity to the Attribute Mapping Utility:

a Select **Entity** →**Add Entity**. A pop-up dialog displays:



Enter Entity Name

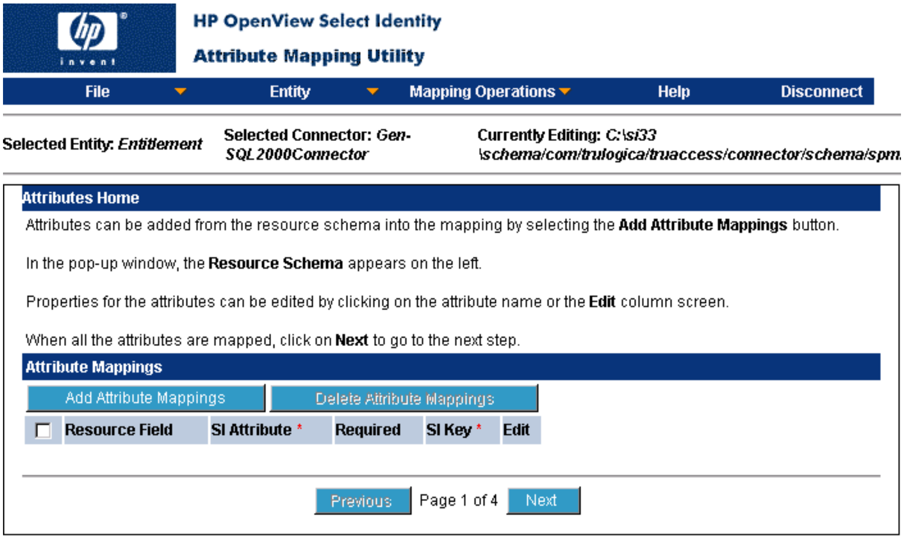
Select Entity to Add

Entity Name To Add

b In the Entity Name To Add field, specify a name for the entitlement entity.

c Click **OK**.

The entitlement entity is selected in the utility window.



HP OpenView Select Identity
Attribute Mapping Utility

Selected Entity: *Entitlement* Selected Connector: *Gen-SQL2000Connector* Currently Editing: *C:\si33\schema/com/trulogica/truaccess/connector/schema/spm*

Attributes Home

Attributes can be added from the resource schema into the mapping by selecting the **Add Attribute Mappings** button.

In the pop-up window, the **Resource Schema** appears on the left.

Properties for the attributes can be edited by clicking on the attribute name or the **Edit** column screen.

When all the attributes are mapped, click on **Next** to go to the next step.

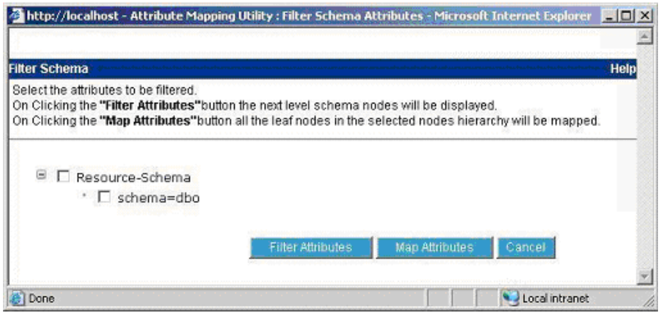
Attribute Mappings

□	Resource Field	SI Attribute *	Required	SI Key *	Edit

Page 1 of 4

2 Map each entitlement attribute defined in the database schema to a Select Identity attribute:

a Click **Add Attribute Mappings** to display the Filter Schema page, which provides a listing of the database schema in a pop-up dialog. This displays the database schema in a hierarchical form.



http://localhost - Attribute Mapping Utility : Filter Schema Attributes - Microsoft Internet Explorer

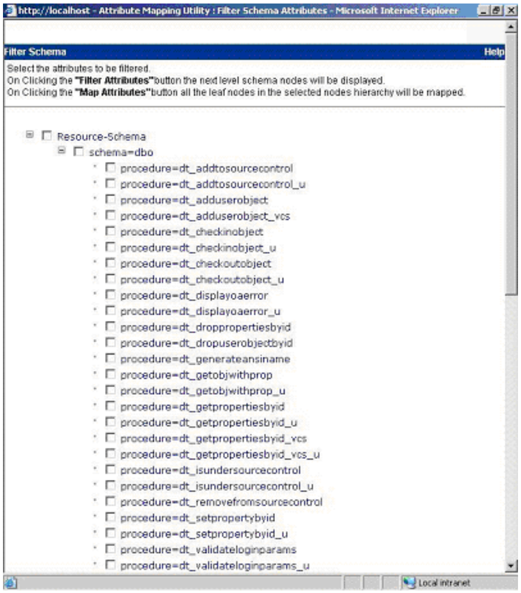
Filter Schema

Select the attributes to be filtered.
On Clicking the **"Filter Attributes"**button the next level schema nodes will be displayed.
On Clicking the **"Map Attributes"**button all the leaf nodes in the selected nodes hierarchy will be mapped.

- □ Resource-Schema
  - □ schema=dbo

From this page, you can filter the tables,views, and stored procedures that you wish to map from the the rest of the database schema. This reduces the schema retrieval time and also provides a better view by showing only the selected schema when mapping attributes.

b To select part of the schema for mapping, select the **Schema** check box and click the **Filter Attributes** button to display the schema. (You can also simply click the **Map Attributes** button to view the entire schema when mapping attributes.)



http://localhost - Attribute Mapping Utility : Filter Schema Attributes - Microsoft Internet Explorer

Filter Schema — Help

Select the attributes to be filtered.
On Clicking the **"Filter Attributes"**button the next level schema nodes will be displayed.
On Clicking the **"Map Attributes"**button all the leaf nodes in the selected nodes hierarchy will be mapped.

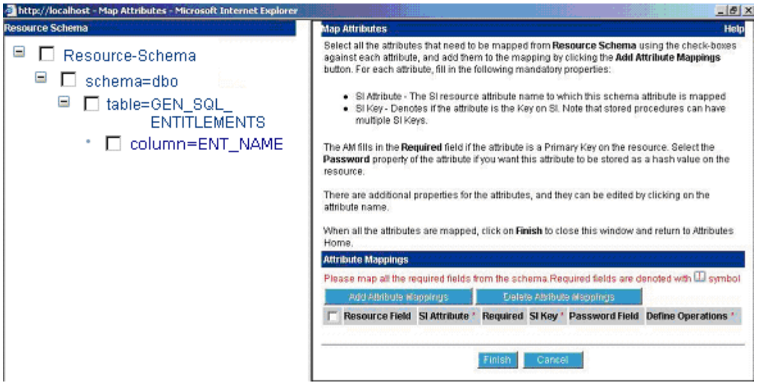
```
⊟ □ Resource-Schema
    ⊟ □ schema=dbo
        · □ procedure=dt_addtosourcecontrol
        · □ procedure=dt_addtosourcecontrol_u
        · □ procedure=dt_adduserobject
        · □ procedure=dt_adduserobject_vcs
        · □ procedure=dt_checkinobject
        · □ procedure=dt_checkinobject_u
        · □ procedure=dt_checkoutobject
        · □ procedure=dt_checkoutobject_u
        · □ procedure=dt_displayoaerror
        · □ procedure=dt_displayoaerror_u
        · □ procedure=dt_droppropertiesbyid
        · □ procedure=dt_dropuserobjectbyid
        · □ procedure=dt_generateansiname
        · □ procedure=dt_getobjwithprop
        · □ procedure=dt_getobjwithprop_u
        · □ procedure=dt_getpropertiesbyid
        · □ procedure=dt_getpropertiesbyid_u
        · □ procedure=dt_getpropertiesbyid_vcs
        · □ procedure=dt_getpropertiesbyid_vcs_u
        · □ procedure=dt_isundersourcecontrol
        · □ procedure=dt_isundersourcecontrol_u
        · □ procedure=dt_removefromsourcecontrol
        · □ procedure=dt_setpropertybyid
        · □ procedure=dt_setpropertybyid_u
        · □ procedure=dt_validateloginparams
        · □ procedure=dt_validateloginparams_u
```

The database schema is shown on the Filter Schema page in a tree that you can expand .

Here is an explanation of the tree:

```
Database Name
    |
    | - Schema Names
    |     | - Table Names
    |     | - Procedure Names
    |     |  - View Names
```

c Select the entitlement tables that you wish to map. After you select tables, click on the **Map Attributes** button. This displays the Map Attributes window, which displays the expanded schema items selected on Filter Schema page.


http://localhost - Map Attributes - Microsoft Internet Explorer
Resource Schema
⊟ □ Resource-Schema
⊟ □ schema=dbo
⊟ □ table=GEN_SQL_ENTITLEMENTS
□ column=ENT_NAME
Map Attributes
Help

Select all the attributes that need to be mapped from **Resource Schema** using the check-boxes against each attribute, and add them to the mapping by clicking the **Add Attribute Mappings** button. For each attribute, fill in the following mandatory properties:

- SI Attribute - The SI resource attribute name to which this schema attribute is mapped
- SI Key - Denotes if the attribute is the Key on SI. Note that stored procedures can have multiple SI Keys.

The AM fills in the **Required** field if the attribute is a Primary Key on the resource. Select the **Password** property of the attribute if you want this attribute to be stored as a hash value on the resource.

There are additional properties for the attributes, and they can be edited by clicking on the attribute name.

When all the attributes are mapped, click on **Finish** to close this window and return to Attributes Home.

Attribute Mappings

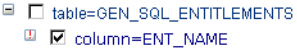
Please map all the required fields from the schema.Required fields are denoted with ! symbol

□	Resource Field	SI Attribute *	Required	SI Key *	Password Field	Define Operations *

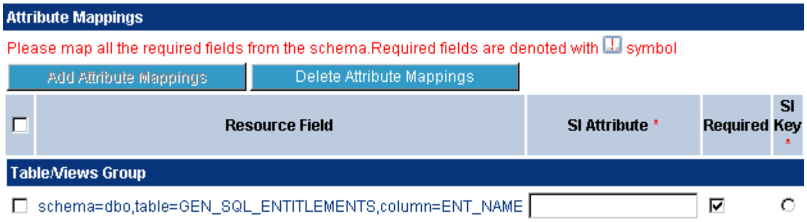
Here is the explanation of the tree:

```
Database Name
    |
    | - Schema Names
    |     | - Table Names
    |     |     | - Column Names
    |     | - Procedure Names
    |     |     | - In Parameters
    |     |     | - Out Parameters
```

d In the left side of the window, select the database column that you would like to map to a Select Identity attribute:



Then, click the **Add Attribute Mappings** button. The selected column is added to the table:


Attribute Mappings

Please map all the required fields from the schema.Required fields are denoted with ! symbol

□	Resource Field	SI Attribute *	Required	SI Key *
Table/Views Group				
□	schema=dbo,table=GEN_SQL_ENTITLEMENTS,column=ENT_NAME		☑	○

e In the SI Attribute field, enter the name of the Select Identity attribute to which you want to map the selected database column. For this scenario, map the ENT_NAME column, enter **`Entitlements`** in the SI Attribute field.

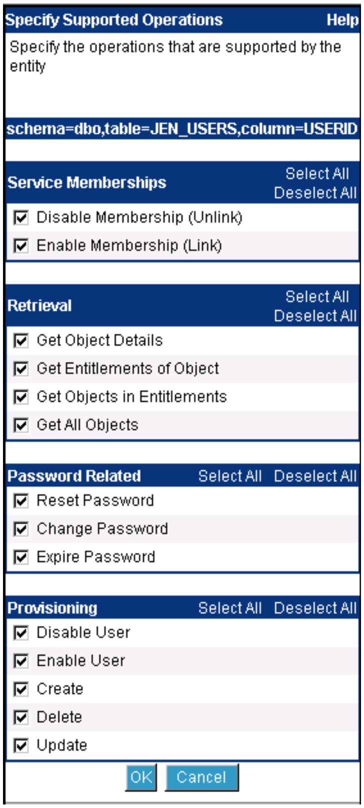
The **Required** option is selected if the chosen database column is the schema's primary key.

f Select the **SI Key** option if the specified attribute is a key field in Select Identity.

g If you wish to encrypt the value of the attribute in the database, such as if the attribute stores a password, select the **Password Field** option.

Typically, attributes of an entitlement or group entity do not require this password behavior. Hence, this option can be ignored unless required.

h If you wish to define the actions that the connector can perform for the attribute (not the entire entity), click the **Define Attribute Operations** button and select the allowed operations. A pop-up window displays in which you can select the operations then click **OK**.



Specify Supported Operations Help

Specify the operations that are supported by the entity

schema=dbo,table=JEN_USERS,column=USERID

Service Memberships Select All Deselect All

☑ Disable Membership (Unlink)

☑ Enable Membership (Link)

Retrieval Select All Deselect All

☑ Get Object Details

☑ Get Entitlements of Object

☑ Get Objects in Entitlements

☑ Get All Objects

Password Related Select All Deselect All

☑ Reset Password

☑ Change Password

☑ Expire Password

Provisioning Select All Deselect All

☑ Disable User

☑ Enable User

☑ Create

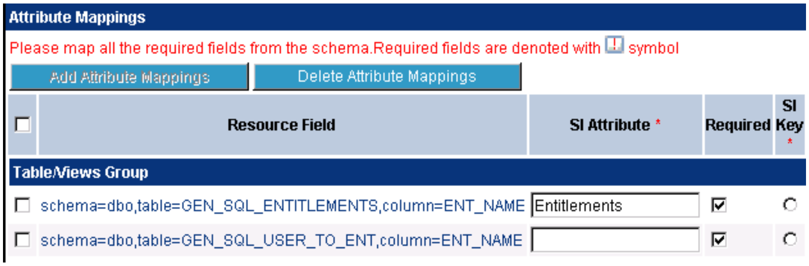
☑ Delete

☑ Update

i Repeat Step d through Step h for additional entitlement columns you wish to map to Select Identity attributes. For this scenario, only the ENT_NAME column is mapped to a Select Identity attribute.

3 *If a mapping table is defined in the database schema:*
Map the entitlement column in the mapping table:

a Select the entitlement column in the mapping table and click **Add Attribute Mappings**. The selected column is added to the list of mapped columns and attributes:



Please map all the required fields from the schema.Required fields are denoted with ! symbol

□	Resource Field	SI Attribute *	Required	SI Key *
Table/Views Group				
□	schema=dbo,table=GEN_SQL_ENTITLEMENTS,column=ENT_NAME	Entitlements	☑	○
□	schema=dbo,table=GEN_SQL_USER_TO_ENT,column=ENT_NAME		☑	○

b In the SI Attribute field, enter an arbitrary value. You simply need to provide a value to enable you to include this column in the mappings.

4 *If an entitlements table (in addition to a user table) is defined in the database schema:*
Map the member column in the entitlements table to the entitlement column in the user table:

a Select the member column in the entitlement table and click **Add Attribute Mappings**. The selected column is added to the list of mapped columns and attributes.

b In the SI Attribute field, enter an arbitrary value. You simply need to provide a value to enable you to include this column in the mappings.

5 Click **Finish** at the bottom of the Map Attributes window.

6 If you have not done so, return to Step 6 on page 51 to define the relationship between the user entity and the entitlement entity.

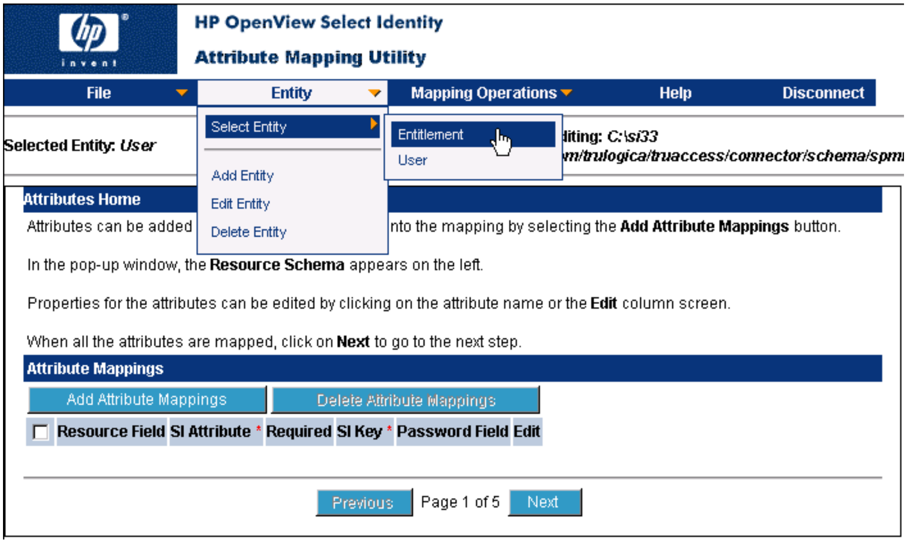
7 Select **File →Save Mapping File** to generate the XML and XSL files in the location specified in the Base Directory field when you connected.

If you wish to download the XML and XSL files to the local system, select **File →Download Mapping File** or **File →Download Reverse Synchronization File**. To disconnect from the database or log out of the utility, click **Disconnect**.

Provisioning Entitlements in the Database

The Attribute Mapping Utility enables you to provision entitlements directly in to the database. Complete the following steps to provision entitlements using the utility:

1 Select the entitlement entity.



Selected Entity: *User*

iting: C:\si33
m/trulogica/truaccess/connector/schema/spml

Attributes Home

Attributes can be added nto the mapping by selecting the **Add Attribute Mappings** button.

In the pop-up window, the **Resource Schema** appears on the left.

Properties for the attributes can be edited by clicking on the attribute name or the **Edit** column screen.

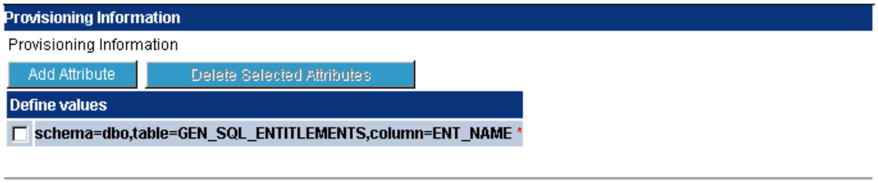
When all the attributes are mapped, click on **Next** to go to the next step.

Attribute Mappings

□	Resource Field	SI Attribute *	Required	SI Key *	Password Field	Edit

Page 1 of 5

2 Select **Provision Entity** from the Mapping Operations menu. The Provisioning Information section displays and lists the attributes that are mapped for the entity.



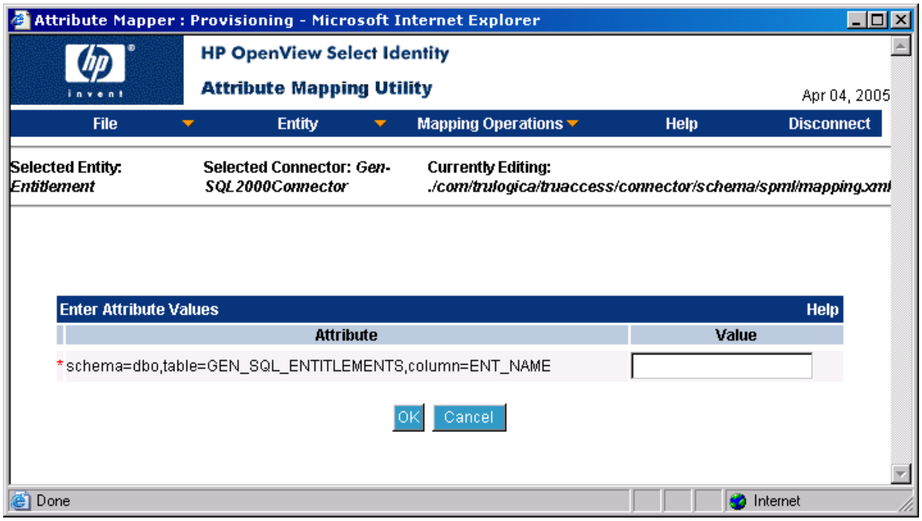
Provisioning Information

Provisioning Information

Define values

□ schema=dbo,table=GEN_SQL_ENTITLEMENTS,column=ENT_NAME *

3 Click **Add Attribute** to modify the values assigned to the attributes. The following window displays:



Attribute Mapper : Provisioning - Microsoft Internet Explorer
HP OpenView Select Identity
Attribute Mapping Utility
Apr 04, 2005
File
Entity
Mapping Operations
Help
Disconnect

Selected Entity: *Entitlement*

Selected Connector: *Gen-SQL2000Connector*

Currently Editing: *./com/trulogica/truaccess/connector/schema/spml/mapping.xml*

Enter Attribute Values — Help

Attribute	Value
*schema=dbo,table=GEN_SQL_ENTITLEMENTS,column=ENT_NAME	

4 Change or enter a value in each Value field for the attributes you wish to provision in the database.

5 Click **OK** when you are done. The value is provisioned in the database.

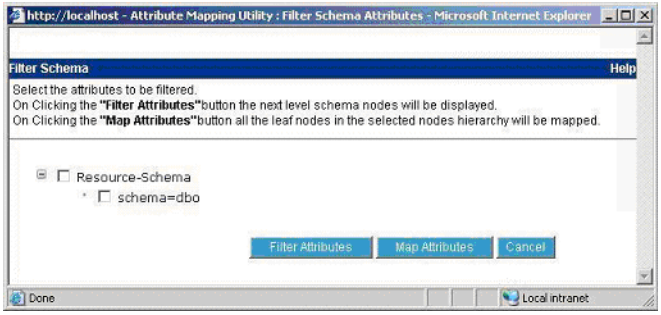
Mapping Stored Procedure Parameters

The following procedure describes how to map stored procedure parameters to Select Identity attributes. When you finish defining the mappings, the XML mapping file is generated in the directory specified in the Base Directory field when you logged in to the Attribute Mapping Utility.

➤ These steps assume that you are logged in to the Attribute Mapping Utility and connected to the database.

Map each parameter for the stored procedure defined in the database schema, as follows:

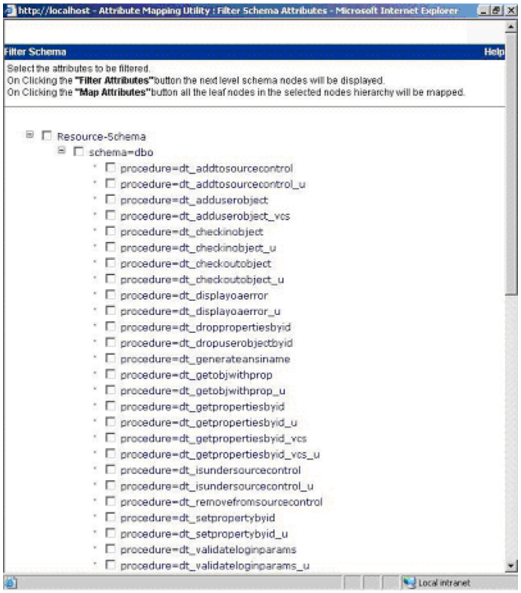
1 Click **Add Attribute Mappings** to display the Filter Schema page, which provides a listing of the database schema in a pop-up dialog. This displays the database schema in a hierarchical form.


http://localhost - Attribute Mapping Utility : Filter Schema Attributes - Microsoft Internet Explorer
Filter Schema
Help
Filter Attributes
Map Attributes
Cancel
Done
Local intranet

Select the attributes to be filtered.
On Clicking the **"Filter Attributes"** button the next level schema nodes will be displayed.
On Clicking the **"Map Attributes"** button all the leaf nodes in the selected nodes hierarchy will be mapped.

- ⊟ □ Resource-Schema
  - □ schema=dbo

2 To select part of the schema for mapping, select the **Schema** check box and click the **Filter Attributes** button to display the schema. (You can also simply click the **Map Attributes** button to view the entire schema when mapping attributes.)


http://localhost - Attribute Mapping Utility : Filter Schema Attributes - Microsoft Internet Explorer
Filter Schema
Help
Local intranet

Select the attributes to be filtered.
On Clicking the **"Filter Attributes"** button the next level schema nodes will be displayed.
On Clicking the **"Map Attributes"** button all the leaf nodes in the selected nodes hierarchy will be mapped.

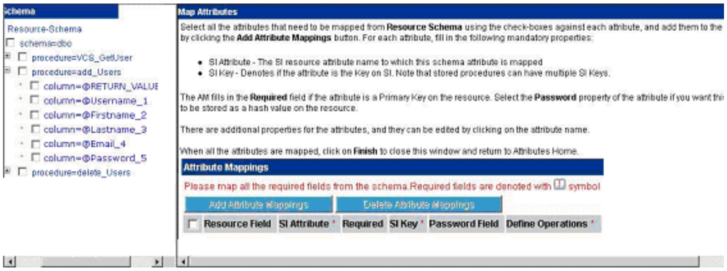
- ⊟ □ Resource-Schema
  - ⊟ □ schema=dbo
    - □ procedure=dt_addtosourcecontrol
    - □ procedure=dt_addtosourcecontrol_u
    - □ procedure=dt_adduserobject
    - □ procedure=dt_adduserobject_vcs
    - □ procedure=dt_checkinobject
    - □ procedure=dt_checkinobject_u
    - □ procedure=dt_checkoutobject
    - □ procedure=dt_checkoutobject_u
    - □ procedure=dt_displayoaerror
    - □ procedure=dt_displayoaerror_u
    - □ procedure=dt_droppropertiesbyid
    - □ procedure=dt_dropuserobjectbyid
    - □ procedure=dt_generateansiname
    - □ procedure=dt_getobjwithprop
    - □ procedure=dt_getobjwithprop_u
    - □ procedure=dt_getpropertiesbyid
    - □ procedure=dt_getpropertiesbyid_u
    - □ procedure=dt_getpropertiesbyid_vcs
    - □ procedure=dt_getpropertiesbyid_vcs_u
    - □ procedure=dt_isundersourcecontrol
    - □ procedure=dt_isundersourcecontrol_u
    - □ procedure=dt_removefromsourcecontrol
    - □ procedure=dt_setpropertybyid
    - □ procedure=dt_setpropertybyid_u
    - □ procedure=dt_validateloginparams
    - □ procedure=dt_validateloginparams_u

The database schema is shown on the Filter Schema page in a tree that you can expand .

Here is an explanation of the tree:

```
Database Name
    |
    | - Schema Names
    |       | - Table Names
    |       | - Procedure Names
    |       |  - View Names
```

c Select the stored procedure tables that you wish to map. After you select tables, click on the **Map Attributes** button. This displays the Map Attributes window, which displays the expanded schema items selected on Filter Schema page.



Schema

Resource-Schema

□ schema=dbo

□ procedure=VCS_GetUser

□ procedure=add_Users

□ column=@RETURN_VALUE

□ column=@Username_1

□ column=@Firstname_2

□ column=@Lastname_3

□ column=@Email_4

□ column=@Password_5

□ procedure=delete_Users

Map Attributes

Select all the attributes that need to be mapped from **Resource Schema** using the check-boxes against each attribute, and add them to the by clicking the **Add Attribute Mappings** button. For each attribute, fill in the following mandatory properties:

- SI Attribute - The SI resource attribute name to which this schema attribute is mapped
- SI Key - Denotes if the attribute is the Key on SI. Note that stored procedures can have multiple SI Keys.

The AM fills in the **Required** field if the attribute is a Primary Key on the resource. Select the **Password** property of the attribute if you want thi to be stored as a hash value on the resource.

There are additional properties for the attributes, and they can be edited by clicking on the attribute name.

When all the attributes are mapped, click on **Finish** to close this window and return to Attributes Home.

Attribute Mappings

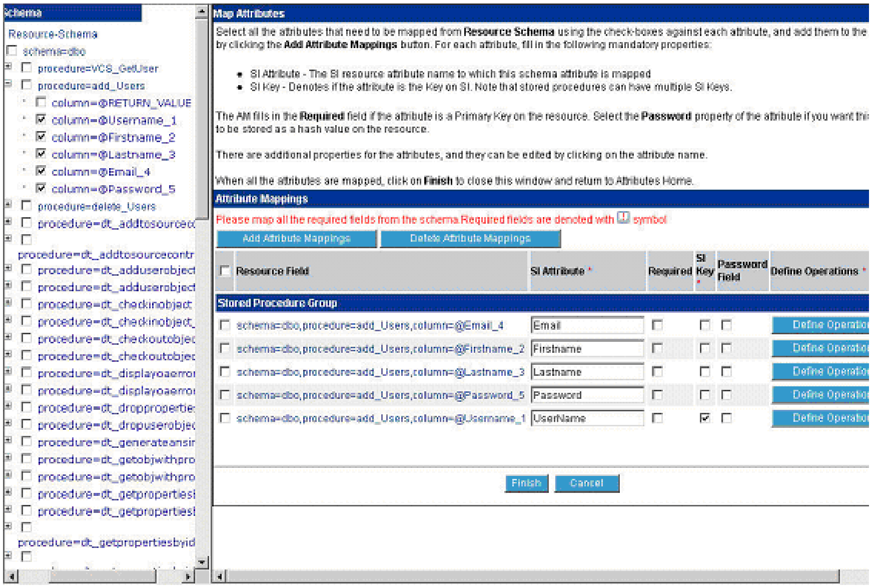
Please map all the required fields from the schema.Required fields are denoted with ☐ symbol

□	Resource Field	SI Attribute *	Required	SI Key *	Password Field	Define Operations *

Here is the explanation of the tree:

```
Database Name
    |
    | - Schema Names
    |       | - Table Names
    |       |       | - Column Names
    |       | - Procedure Names
    |       |       | - In Parameters
    |       |       | - Out Parameters
```

3 From the left side of the window, select the stored procedure parameters that you would like to map to Select Identity attributes. In the example below, all of the parameters for the addUser stored procedure (except the RETURN VALUE) are mapped:



Resource-Schema

- □ schema=dbo
  - □ procedure=VCS_GetUser
  - □ procedure=add_Users
    - □ column=@RETURN_VALUE
    - ☑ column=@Username_1
    - ☑ column=@Firstname_2
    - ☑ column=@Lastname_3
    - ☑ column=@Email_4
    - ☑ column=@Password_5
  - □ procedure=delete_Users
  - □ procedure=dt_addtosourcec
  - □ procedure=dt_addtosourcecontr
  - □ procedure=dt_adduserobject
  - □ procedure=dt_adduserobject
  - □ procedure=dt_checkinobject
  - □ procedure=dt_checkinobject_
  - □ procedure=dt_checkoutobjec
  - □ procedure=dt_checkoutobjec
  - □ procedure=dt_displayoaerror
  - □ procedure=dt_displayoaerror
  - □ procedure=dt_droppropertie
  - □ procedure=dt_dropuserobjec
  - □ procedure=dt_generateansir
  - □ procedure=dt_getobjwithpro
  - □ procedure=dt_getobjwithpro
  - □ procedure=dt_getpropertiesl
  - □ procedure=dt_getpropertiesl
  - □ procedure=dt_getpropertiesbyid
  - □

Select all the attributes that need to be mapped from **Resource Schema** using the check-boxes against each attribute, and add them to the by clicking the **Add Attribute Mappings** button. For each attribute, fill in the following mandatory properties:

- SI Attribute - The SI resource attribute name to which this schema attribute is mapped
- SI Key - Denotes if the attribute is the Key on SI. Note that stored procedures can have multiple SI Keys.

The AM fills in the **Required** field if the attribute is a Primary Key on the resource. Select the **Password** property of the attribute if you want thi to be stored as a hash value on the resource.

There are additional properties for the attributes, and they can be edited by clicking on the attribute name.

When all the attributes are mapped, click on **Finish** to close this window and return to Attributes Home.

Attribute Mappings

Please map all the required fields from the schema.Required fields are denoted with ! symbol

□	Resource Field	SI Attribute *	Required	SI Key *	Password Field	Define Operations *
Stored Procedure Group						
□	schema=dbo,procedure=add_Users,column=@Email_4	Email	□	□	□	Define Operatio
□	schema=dbo,procedure=add_Users,column=@Firstname_2	Firstname	□	□	□	Define Operatio
□	schema=dbo,procedure=add_Users,column=@Lastname_3	Lastname	□	□	□	Define Operatio
□	schema=dbo,procedure=add_Users,column=@Password_5	Password	□	□	□	Define Operatio
□	schema=dbo,procedure=add_Users,column=@Username_1	UserName	□	☑	□	Define Operatio

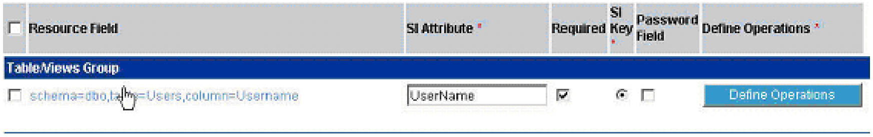
4 Map the parameters of the modifyUser and deleteUser stored procedures.

⚠ Be sure to specify the Select Identity attributes exactly as they appear in the Select Identity client.

Note that similar parameters may occur multiple times in the mapping, such as in this scenario; the Username parameter is requred by all stored procedures. It is recommended that all Username parameters and table columns specify the same Select Identity attribute name.

5 Map the key column of the table into which the stored procedures are provisioning data. This is the column used to uniquely identify a user in the database tables and is used as a field to verify that the user is available to the connector.

In this scenario, all the stored procedures use the Username column of the Users table as the key field. Hence, this is mapped in the utility.



□	Resource Field	SI Attribute *	Required	SI Key *	Password Field	Define Operations *
Table/Views Group						
□	schema=dbo,table=Users,column=Username	UserName	☑	●	□	Define Operations

6 Provide the Select Identity key information for the mapped attributes. There are two cases here:

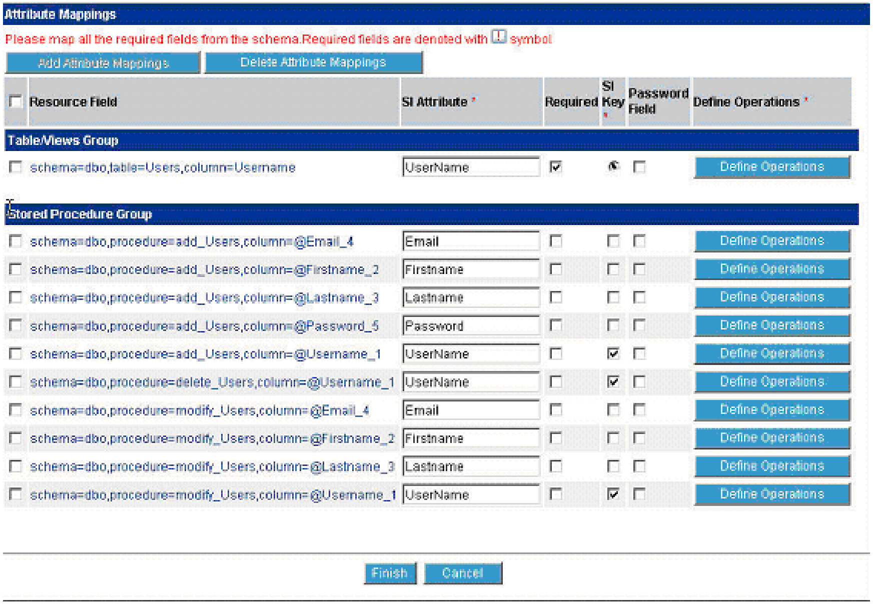
— The table column that is mapped is present in a separate group and the SI Key radio button is available. A radio button is available because only one SI Key attribute is supported for table column-related scenarios. Hence, the table column that is designated as the SI Key is the Select Identity key field.

— The store procedures parameters are grouped separately and the SI Key check box is available for each. Check boxes are available because you may have multiple Select Identity keys for stored procedures. Thus, check the SI Key option for all of the parameters that are analogous to the Select Identity key attribute of the mapped table column.

In the example scenario, the Username column in the Users table is the key field in Select Identity. For this column, the SI Key check box is selected in the Table/View Group.

In the Stored Procedure group, the SI Key check box is selected for all of the parameters that correspond to this table column (mapped to the

Username parameter of adduser, modifyUser, and deleteUser stored procedures).



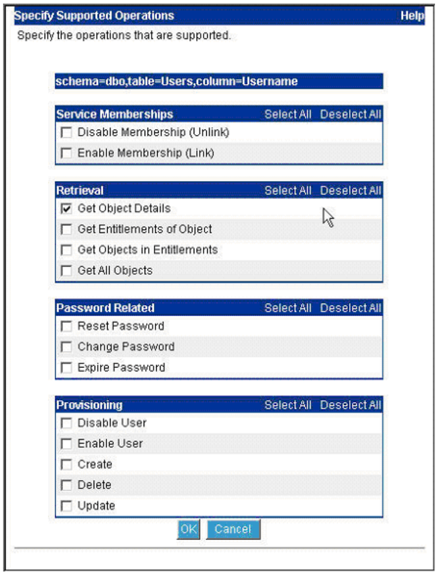
Attribute Mappings

Please map all the required fields from the schema.Required fields are denoted with ! symbol

□	Resource Field	SI Attribute *	Required	SI Key *	Password Field	Define Operations *
	Table/Views Group					
□	schema=dbo,table=Users,column=Username	UserName	☑	●	□	Define Operations
	Stored Procedure Group					
□	schema=dbo,procedure=add_Users,column=@Email_4	Email	□	□	□	Define Operations
□	schema=dbo,procedure=add_Users,column=@Firstname_2	Firstname	□	□	□	Define Operations
□	schema=dbo,procedure=add_Users,column=@Lastname_3	Lastname	□	□	□	Define Operations
□	schema=dbo,procedure=add_Users,column=@Password_5	Password	□	□	□	Define Operations
□	schema=dbo,procedure=add_Users,column=@Username_1	UserName	□	☑	□	Define Operations
□	schema=dbo,procedure=delete_Users,column=@Username_1	UserName	□	☑	□	Define Operations
□	schema=dbo,procedure=modify_Users,column=@Email_4	Email	□	□	□	Define Operations
□	schema=dbo,procedure=modify_Users,column=@Firstname_2	Firstname	□	□	□	Define Operations
□	schema=dbo,procedure=modify_Users,column=@Lastname_3	Lastname	□	□	□	Define Operations
□	schema=dbo,procedure=modify_Users,column=@Username_1	UserName	□	☑	□	Define Operations

7 Click the **Define Operations** button for the mapped table column that is used for verifying that the user exists in the database table. For this

attribute, select only the **Get Object Details** operation (deselect all other operations).



Specify Supported Operations

Specify the operations that are supported.

schema=dbo,table=Users,column=Username

Service Memberships

□ Disable Membership (Unlink)

□ Enable Membership (Link)

Retrieval

☑ Get Object Details

□ Get Entitlements of Object

□ Get Objects in Entitlements

□ Get All Objects

Password Related

□ Reset Password

□ Change Password

□ Expire Password

Provisioning

□ Disable User

□ Enable User

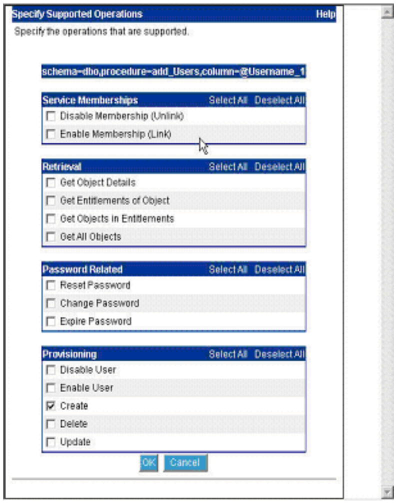
□ Create

□ Delete

□ Update

8 Click **Define Operations** for each of the stored procedure parameters and select only the relevant operations for which the parameter is used.

For example, the Username parameter of the addUser stored procedure will be used when performing Add operations only. Therefore, only the Create operation must be selected for this parameter.



Specify Supported Operations

Specify the operations that are supported.

schema=dbo,procedure=add_Users,column=@Username_1

Service Memberships

□ Disable Membership (Unlink)

□ Enable Membership (Link)

Retrieval

□ Get Object Details

□ Get Entitlements of Object

□ Get Objects in Entitlements

□ Get All Objects

Password Related

□ Reset Password

□ Change Password

□ Expire Password

Provisioning

□ Disable User

□ Enable User

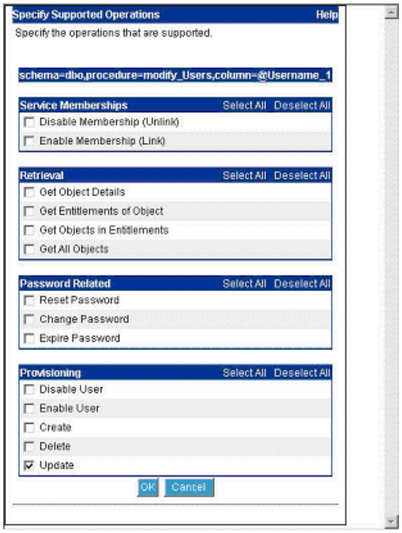
☑ Create

□ Delete

□ Update

Similarly, the Email parameter is required by the addUser and modifyUser procedures. The Email parameter of addUser should be enabled for Create operations only and the Update operation should be enabled for the Email parameter of the modifyUser stored procedure.

The Update operation must be enables for the Username parameter of the modifyUser procedure.



Specify Supported Operations

Specify the operations that are supported.

schema=dbo,procedure=modify_Users,column=@Username_1

Service Memberships

□ Disable Membership (Unlink)
□ Enable Membership (Link)

Retrieval

□ Get Object Details
□ Get Entitlements of Object
□ Get Objects in Entitlements
□ Get All Objects

Password Related

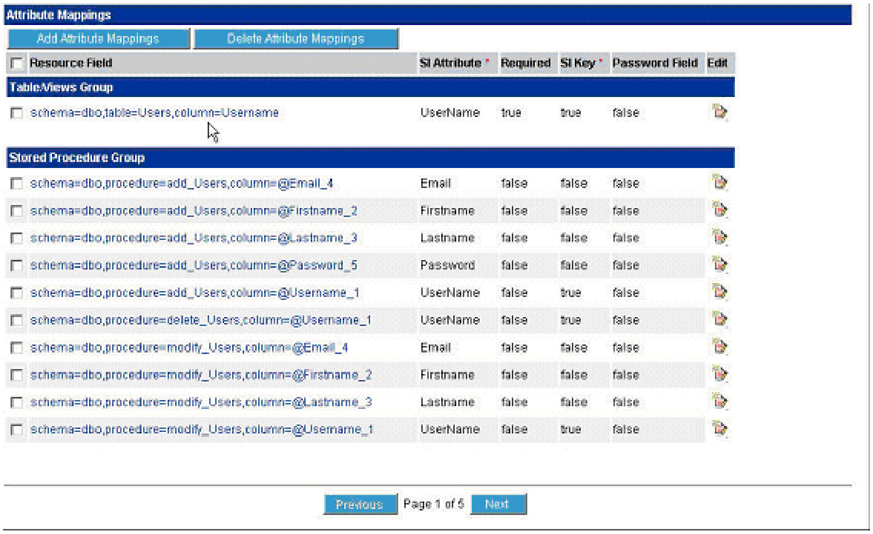
□ Reset Password
□ Change Password
□ Expire Password

Provisioning

□ Disable User
□ Enable User
□ Create
□ Delete
☑ Update

Finally, the Delete operation must be enabled for the Username parameter of the deleteUser procedure.

9 Click on the **Finish** button to return to the Attributes page. This displays the mappings for the table columns and stored procedure parameters. You can click on the **Edit** link for any attribute to modify the details.



Attribute Mappings

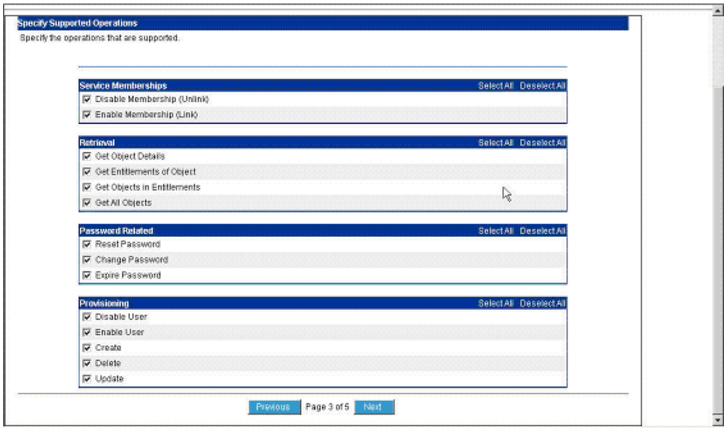
Add Attribute Mappings | Delete Attribute Mappings

□ Resource Field	SI Attribute *	Required	SI Key *	Password Field	Edit
Table/Views Group					
□ schema=dbo,table=Users,column=Username	UserName	true	true	false	✎
Stored Procedure Group					
□ schema=dbo,procedure=add_Users,column=@Email_4	Email	false	false	false	✎
□ schema=dbo,procedure=add_Users,column=@Firstname_2	Firstname	false	false	false	✎
□ schema=dbo,procedure=add_Users,column=@Lastname_3	Lastname	false	false	false	✎
□ schema=dbo,procedure=add_Users,column=@Password_5	Password	false	false	false	✎
□ schema=dbo,procedure=add_Users,column=@Username_1	UserName	false	true	false	✎
□ schema=dbo,procedure=delete_Users,column=@Username_1	UserName	false	true	false	✎
□ schema=dbo,procedure=modify_Users,column=@Email_4	Email	false	false	false	✎
□ schema=dbo,procedure=modify_Users,column=@Firstname_2	Firstname	false	false	false	✎
□ schema=dbo,procedure=modify_Users,column=@Lastname_3	Lastname	false	false	false	✎
□ schema=dbo,procedure=modify_Users,column=@Username_1	UserName	false	true	false	✎

Previous | Page 1 of 5 | Next

10 From the Mapping Operations menu, select **Define Entity Operations**.

11 On the page that displays, select the operations that are required to support the entity.



Specify Supported Operations

Specify the operations that are supported.

Service Memberships — Select All Deselect All

- ☑ Disable Membership (Unlink)
- ☑ Enable Membership (Link)

Retrieval — Select All Deselect All

- ☑ Get Object Details
- ☑ Get Entitlements of Object
- ☑ Get Objects in Entitlements
- ☑ Get All Objects

Password Related — Select All Deselect All

- ☑ Reset Password
- ☑ Change Password
- ☑ Expire Password

Provisioning — Select All Deselect All

- ☑ Disable User
- ☑ Enable User
- ☑ Create
- ☑ Delete
- ☑ Update

Previous | Page 3 of 5 | Next

12 From the File menu, select the **Save Mapping File** option to save the XML file.

A

Usage Scenarios

This appendix provides common usage scenarios for configuring a database schema. Scenarios are organized in the following sections:

- Provisioning Identity Information on page 63
- Synchronizing Changes with Select Identity on page 69

Provisioning Identity Information

The scenarios in this section describe provisioning identity information using user tables, stored procedures, and a combination of tables and stored procedures.

One User Table

A single database table is created and contains all user information. This table has columns for all the user attributes, such as the name, ID, password, email address, and so on.

When the resource is deployed in Select Identity, the connector loads all of the column names from the user table as resource attributes. You can then map Select Identity attributes to the table columns and provision users.

One User Table, One Entitlements Table

A single database table is created and contains all user information. This table has columns for all user attributes, such as the name, ID, password, email address, and so on. In addition, the user table provides a column that contains all of the user’s entitlements (such as memberOf). This is a multi-valued attribute and the mapping could implement it as a CSV, such as "group1, group2, group3".

Another table is created to store the entitlements. This table has columns that define each entitlement, such as the name and description. In addition, this table has a column for listing users belonging to this entitlement (such as members). This is a multi-valued attribute and the mapping could implement it as a CSV, such as "user1, user2, user3, user4". Populate the entitlement table with entitlements that are to be associated and dissociated with the user.

When the resource is deployed in Select Identity, the connector loads all of the column names from the user table as resource attributes. You can then map Select Identity attributes to the table columns and provision users.

Select Identity also retrieves the entitlements during the Service configuration or user creation, and connector loads all entries from the entitlement table.

The users are provisioned in the members column of the database.

For example, if you create a group entity names Entitlement for the entitlement table in the schema, and you provision a user with the administrator entitlement (in Select Identity), the memberOf column of that user will have a value “administrator”. If more than one entitlement is assigned, they are comma separated, as in this example: administrator,guest. Similarly, the user will be provisioned in the members column of the Entitlement table for each entitlement assigned to him or her. If more than one user is given the same administrator entitlement, the members column for the administrator row will have “user1,user2” as its value.

One User Table, One Entitlements Table, One Map Table

A single database table is created and contains all user information. This table has columns for all user attributes, such as the name, ID, password, email address, and so on.

Another table is created to store the entitlements. This table has columns that define each entitlement, such as the name and description. In addition, this table has a column for listing users belonging to this entitlement (such as members). This is a multi-valued attribute and the mapping could implement it as a CSV, such as "user1, user2, user3, user4". Populate the entitlement table with entitlements that are to be associated and dissociated with the user.

A third table is created to store the user-to-entitlement mappings. This table refers to the primary keys (PKs) of user table and entitlements table. This map table has columns such as user name and entitlements.

When the resource is deployed in Select Identity, the connector loads all of the column names from the user table as resource attributes. You can then map Select Identity attributes to the table columns and provision users.

Select Identity also retrieves the entitlements during the Service configuration or user creation, and connector loads all entries from the entitlement table.

If more than one entitlement is assigned to a user, the values are provisioned in separate rows in the map table.

For example, if the user1 user is assigned the administrator and guest entitlements, the map table will have two rows, user1-administrator and user1-guest.

Multiple User Tables

A table is created to store some user information. This table can have columns to store such attributes as user name, ID, password, email address, and so.

Additional tables can be created to store additional user information. These tables can contain columns to store data such as company, department, costcenter, address, phone number, and so on. The tables must have a foreign key constraint against the main user table.

When the resource is deployed in Select Identity, the connector loads all of the column names from the user tables as resource attributes. You can then map Select Identity attributes to the table columns and provision users. Provisioning should populate the user tables.

Two User Tables, One Entitlements Table

A table is created to store some user information. This table can have columns to store such attributes as user name, ID, password, email address, and so. In addition, the user table provides a column that contains all of the user's entitlements (such as memberOf). This is a multi-valued attribute and the mapping could implement it as a CSV, such as "group1, group2, group3".

Another table is created to store additional user information. This table can contain columns to store data such as company, department, costcenter, address, phone number, and so on. This table has a foreign key constraint against the main user table.

A third table is created to store the entitlements. This table has columns that define each entitlement, such as the name and description. In addition, this table has a column for listing users belonging to this entitlement (such as members). This is a multi-valued attribute and the mapping could implement it as a CSV, such as "user1, user2, user3, user4". Populate the entitlement table with entitlements that are to be associated and dissociated with the user.

When the resource is deployed in Select Identity, the connector loads all of the column names from the user tables as resource attributes. You can then map Select Identity attributes to the table columns and provision users.

Select Identity also retrieves the entitlements during the Service configuration or user creation, and connector loads all entries from the entitlement table.

Two User Tables, One Entitlements Table, One Map Table

A table is created to store some user information. This table can have columns to store such attributes as user name, ID, password, email address, and so.

Another table is created to store additional user information. This table can contain columns to store data such as company, department, costcenter, address, phone number, and so on. This table has a foreign key constraint against the main user table.

A third table is created to store the entitlements. This table has columns that define each entitlement, such as the name and description. In addition, this table has a column for listing users belonging to this entitlement (such as members). This is a multi-valued attribute and the mapping could implement it as a CSV, such as "user1, user2, user3, user4". Populate the entitlement table with entitlements that are to be associated and dissociated with the user.

A fourth table is created to store the user-to-entitlement mappings. This table refers to the primary keys (PKs) of user table and entitlements table. This map table has columns such as user name and entitlements.

When the resource is deployed in Select Identity, the connector loads all of the column names from the user tables as resource attributes. You can then map Select Identity attributes to the table columns and provision users.

Select Identity also retrieves the entitlements during the Service configuration or user creation, and connector loads all entries from the entitlement table.

Multiple User Tables, Multiple Entitlement Tables

Multiple user tables and entitlement tables can be created, where the user information is store in several tables. Each entitlement table has a specific type of entitlements. For example, there could be a table for user groups, one for roles, and one for access control levels.

Because there is more than one entitlement table, you must create more than one group entity for the connector’s mapping file (using that attribute mapping utility). For example, if the Roles and ACL entitlement tables exist in schema, you must create two entities to map these two entitlement tables.

Single Stored Procedure

A stored procedure is created in the database, which will be called by the connector for all provisioning operations. When mapping attributes (such as using the attribute mapping utility), map all of the Select Identity user attributes to the arguments of the stored procedure. When the connector provisions users, the stored procedure is called with the argument values as attributes.

Multiple Stored Procedures

A separate stored procedure is created in the database for each of the user provisioning operations, such as to add a user, modify a user, delete a user, get a user, enable a user, disable a user, reset a password, and so on. When mapping attributes (such as using the attribute mapping utility), map all of the Select Identity operations to the stored procedures. Then, map all of the Select Identity user attributes to the arguments of the stored procedures. When the connector provisions users, the stored procedures are called with the argument values as attributes.

A mapping file for this scenario must map the parameters of the stored procedures and it must specify the attribute level operations (Define Operation) for each parameter. For example, the UserID parameter of the createUser stored procedure will have “create” in its defined operations because the parameter should be used only for the Add User operation.

Stored Procedure for Attributes

Certain attribute information must be encrypted before being stored in the database. The connector passes data to the stored procedure that it calls. You can create one or more stored procedures to be called for encryption of attribute information. Then, map the attributes to the stored procedures.

Tables and Stored Procedures

There may be scenarios in which a user is provisioned using a combination of updating table and invoking stored procedures in the database. For example, a user could be added to tables but a stored procedure is invoked to enable or disable the user. Review the table and stored procedure scenarios above to understand the necessary configuration.

Synchronizing Changes with Select Identity

Synchronizing identity changes is supported only when the agent is deployed on the resource. Database-specific triggers must be installed and configured. The agent can then send changes made to the information stored in user tables to Select Identity by sending SPML requests to the Web Service URL, for reconciliation. The following changes can be sent to Select Identity:

- Add new entries in the user table(s)
- Update entries:
  - Change attribute information (column data)
  - Add or delete entitlements
- Delete entries