HP OpenView Select Identity

Connector for Microsoft SQL Server 2000 Administration

Installation and Configuration Guide

Connector Version: 1.1
Select Identity Version: 3.3

April 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/).

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.
- Java Service Wrapper, Copyright © 1999, 2004 Tanuki Software.
- Copyright © 2001 Silver Egg Technology.
- This product includes software developed by Teodor Danciu (http://jasperreports.sourceforge.net). Portions Copyright © 2001-2004 Teodor Danciu (teodord@users.sourceforge.net). All rights reserved.
- Portions Copyright 1994-2004 Sun Microsystems, Inc. All Rights Reserved.

- This product includes software developed by the Waveset Technologies, Inc. (www.waveset.com). Portions Copyright © 2003 Waveset Technologies, Inc. 6034 West Courtyard Drive, Suite 210, Austin, Texas 78730. All rights reserved.
- Portions Copyright © 2001-2004, Gaudenz Alder. All rights reserved.

Trademark Notices

HP OpenView Select Identity is a trademark of Hewlett-Packard Development Company, L.P.

Microsoft, Windows, the Windows logo, and SQL Server are trademarks or registered trademarks of Microsoft Corporation.

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

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

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

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://passport.hp.com/hpp2/newuser.do

contents

1

Installing the Connector

The Microsoft SQL Server 2000 Administration connector — hereafter referred to as the SQL Server Admin connector — enables HP OpenView Select Identity to administer the database server by provisioning database user information in system schemas. The connector is a two-way connector. Changes made to system user attributes in the database can also be propagated back to Select Identity.

Three configurations are supported for the SQL Server Admin connector:

- Agent-based
  In this configuration, the connector communicates with an agent that resides on the database server; the agent uses a JDBC 2.0 compliant driver to communicate with the database. The agent can also push changes made in SQL Server to the Select Identity database (this is called **reverse synchronization** and explained later).
- Agentless using a JDBC data source
  In this configuration, the connector communicates the database directly through JDBC calls. Be sure to create or identify a JDBC data source (and underlying connection pool) on the Select Identity server that can connect to the target SQL Server database.
- Agentless using a JDBC driver
  The connector communicates the database using a JDBC 2.0 compliant driver; no agent is installed on the database server.

The SQL Server Admin connector is packaged in the following files and folders, which are located on the Select Identity Connector CD:

- `MS SQL Server - Admin/Admin-SQL2000-Connector.rar` — The binaries for the connector
- `MS SQL Server - Admin/Admin SQL Schema.zip` — The mapping files (`Admin_SQL.xml` and `Admin_SQL.xsl`) for the connector
- `MS SQL Server - Admin/Agent Installers/SQL-Admin-AgentInstaller-Win.zip` — A ZIP file that contains the installation executable for the connector agent
- `MS SQL Server - Admin/Agent Installers/SQL-Admin-AgentInstaller-Unix.tar` — A TAR file that contains the installation executable for the connector agent
- `MS SQL Server - Admin/Manual Agent/SQL-Admin-Agent-Win.zip` — A ZIP file that contains agent binaries and files (for manual installation)
- `MS SQL Server - Admin/Manual Agent/SQL-Admin-Agent-Unix.tar` — A TAR file that contains agent binaries and files (for manual installation)

Operations Supported by the Connector

The SQL Server Admin connector is intended for use in a wide variety of usage scenarios. Specifically, it can perform the following operations on the SQL Server system:

- Add, update, and remove users
- Retrieve user attributes
- Enable and disable users
- Verify a user's existence
- Change user passwords
- Reset user passwords
- Retrieve all entitlements

- Retrieve a list of supported user attributes
- Assign and unassign entitlements to and from users

When you provision a user using this connector, be sure to grant administrative privileges to the user for the target database.

In addition, the connector's agent can send user changes made in SQL Server to Select Identity. When changes are pushed from the agent to the Select Identity server, this is referred to as **reverse synchronization**. Specifically, the agent can add, modify, and delete users in Select Identity based on user additions, modifications, and deletions in SQL Server.

When a user is added, modified, or deleted in the database, reverse notification tables capture the changes. The agent's reverse synchronization component then sends the changes to Select Identity's Web Service in SPML. If an error occurs during reverse synchronization, the agent stops the operation (without affecting the connector's operations).

Additional steps are required to configure the agent for reverse synchronization. (Note that installing and configuring the agent is mandatory in order for the connector to support reverse synchronization.)

System Requirements

The SQL Server Admin connector is supported in the following environment:

Select Identity Version	Application Server	Database
3.0.2	WebLogic 8.1.2 on Windows 2003	SQL Server 2000
3.3	WebLogic 8.1.4 on Windows 2003	SQL Server 2000
	WebLogic 8.1.4 on Red Hat Enterprise Linux 3.0	SQL Server 2000

For Select Identity 3.0.2, the SQL Server Admin connector is supported with Microsoft SQL Server 2000 running on Windows 2000, Windows 2003, and Windows XP. For 3.3, the connector is supported with Microsoft SQL Server 2000 running on Windows 2003.

Deploying on the Web Application Server

To install the SQL Server Admin connector on the Select Identity server, complete these steps:

1. Create a subdirectory in the Select Identity home directory where the connector's RAR file will reside. For example, you could create the `C:\Select_Identity\connectors` folder on Windows. (A connector subdirectory may already exist.)
2. Copy the `Admin-SQL2000-Connector.rar` file from the Select Identity Connector CD to the connector subdirectory.
3. Create a schema subdirectory in the Select Identity home directory where the connector's mapping files will reside. For example, you could create the `C:\Select_Identity\schema` folder. (This subdirectory may already exist.)
4. Extract the contents of the `Admin SQL Schema.zip` file (on the Select Identity Connector CD) to the schema subdirectory. The XSL file is extracted into the `Admin SQL Schema` subdirectory, and the XML file is extracted into the `Admin SQL Schema/com/trulogica/truacess/connector/schema/spml` subdirectory.
5. Copy the JDBC 2.0 compliant driver to the application server. For SQL Server, you must copy the JDBC driver files (`msbase.jar`, `mssqlserver.jar`, and `msutil.jar`). Obtain these files from your database administrator.
6. Add the JDBC driver and schema subdirectory to the application server's class path, such as by editing the `myStartWL.cmd` (on Windows) or `myStartWL.sh` (on UNIX) file.
7. If deploying the connector on WebLogic, complete the following steps.
   a. Start the application server if it is not currently running.
   b. Log on to the WebLogic Server Console.
   c. Navigate to ***My_domain*** → **Deployments** → **Connector Modules**.
   d. Click **Deploy a New Connector Module**.
   e. Locate and select the `Admin-SQL2000-Connector.rar` file from the list. It is stored in the connector subdirectory.
   f. Click **Target Module**.

g Select the **My Server** (your server instance) check box.

h Click **Continue**. Review your settings.

i Keep all default settings and click **Deploy**. The Status of Last Action column should display Success.

8 Modify the mapping file, if necessary. This file is described in detail in Understanding the Mapping File on page 15.

9 To configure reverse synchronization on the server, you must create an XSL file based on the XML mapping file. The XSL file maps user attributes on SQL Server to attributes in Select Identity. See Understanding the Mapping Files on page 31 for more information.

Note that the agent must be installed and configured for the SQL Server Admin connector to support reverse synchronization.

After installing the connector, refer to Configuring the Connector on page 25 for information about registering and configuring this connector in Select Identity.

Installing the Agent on the Database Server

After you install the SQL Server Admin connector on the Select Identity server, you can install the agent on the database server. This is optional; the connector can provision users in SQL Server without the agent. However, the agent enables you to send data back to Select Identity (reverse synchronization).

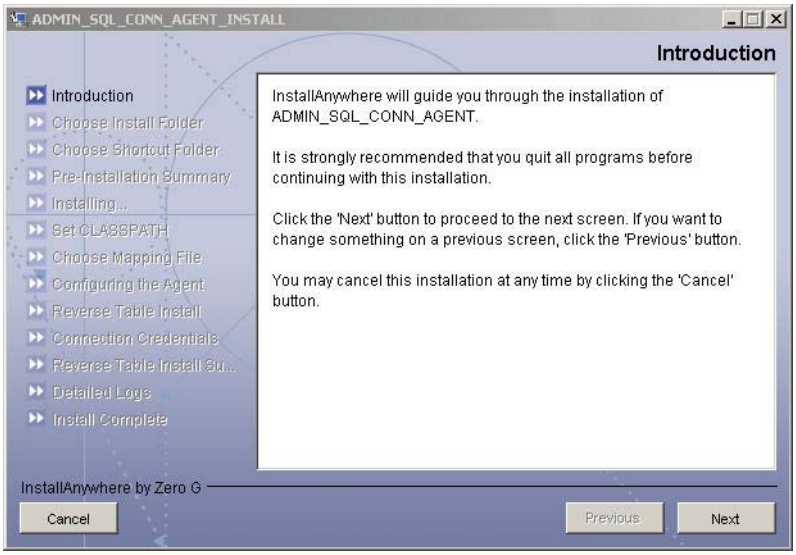
You can install the agent using the installation wizard or by manually copying files to the server.

You must copy the mapping files from the Select Identity server to the system where you will install the agent (on the database server). The agent installation requires that the mapping files are available on the local system.

Installation Using the Wizard on Windows

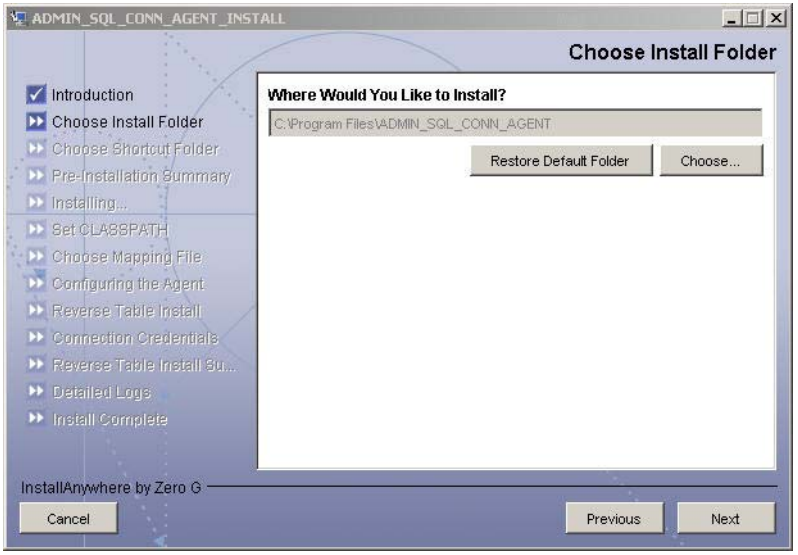
Complete the following steps to run the installation wizard, which installs the agent on Windows:

1. Extract the contents of the `SQL-Admin-AgentInstaller-Win.zip` file, which is located in the `Agent Installers` directory on the CD.
2. Run `install.exe` , which is located in the *`target_dir`*`\CDROM_Installers\Windows\Disk1\InstData\NoVM`. The following dialog displays:



3. Click **Next** to proceed.

4 Specify an installation directory on the Choose Install Folder dialog then click **Next**. By default, the agent is installed in `C:\Program Files\ADMIN_SQL_CONN_AGENT`.

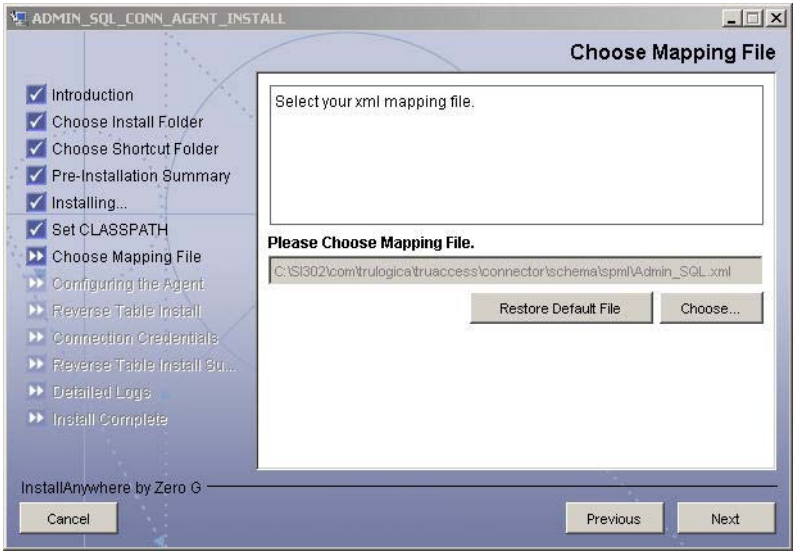


5 Select the location(s) where the product icons will be installed, then click **Next**.

6 Verify the pre-installation summary. If you wish to make changes, click **Previous** and edit the chosen options. To install the agent, click **Install**.

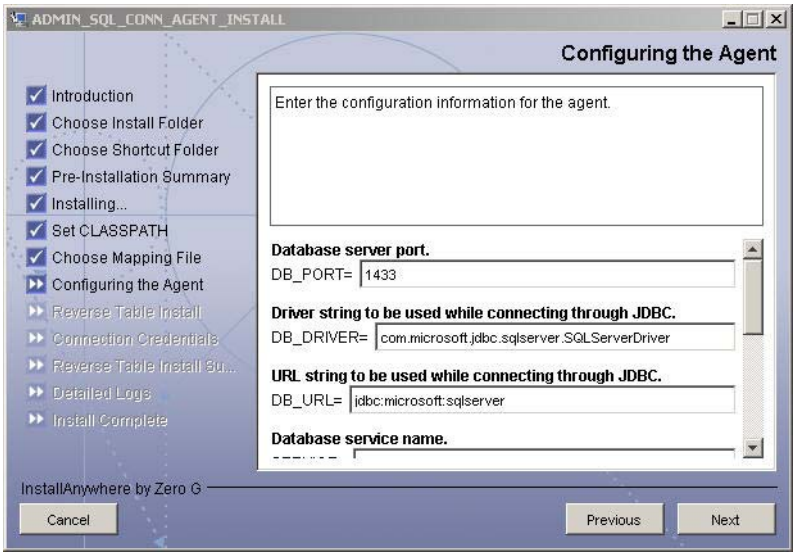
7 On the Set CLASSPATH dialog, click **Next** after you verify that the database driver files (`msbase.jar`, `mssqlserver.jar`, and `msutil.jar`) are in the database server's system CLASSPATH.

8 Click the **Choose** button and select the `Admin_SQL.xml` mapping file that was copied from the Select Identity server. This will copy the mapping file to the *`install_dir`*`/conf/com/trulogica/truaccess/connector/schema/spml` directory, where *install_dir* is the installation folder selected in Step 4 above.



Then, click **Next**.

9 On the Configuring the Agent dialog, specify the requested configuration information:



The following provides an explanation of the configuration options:

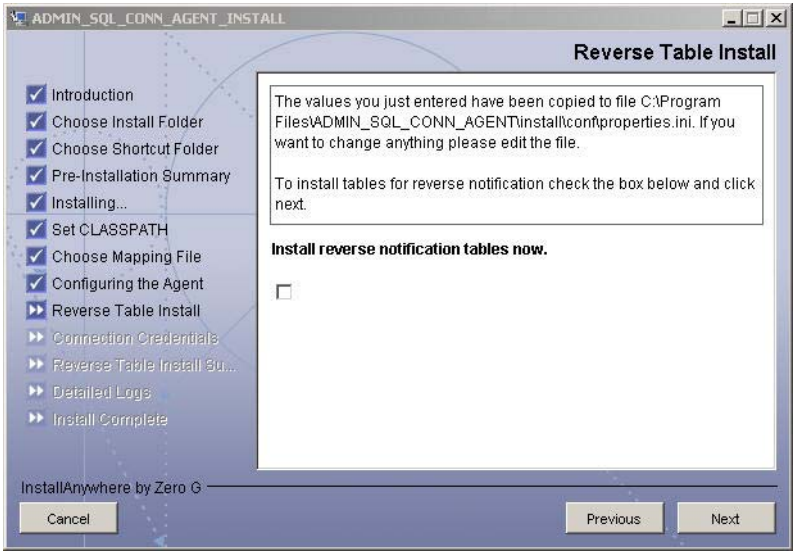
Option	Description	Example Value
DB_PORT	The port on which the database server is listening.	1433
DB_DRIVER	The JDBC driver for the database connection.	com.microsoft.jdbc.sqlserver.SQLServerDriver
DB_URL	The JDBC URL string used for the database communication.	jdbc:microsoft:sqlserver
SERVICE	The database name.	SI_DB
CONCERO_SERVER_URL	The URL of the Select Identity Web Service.	http://*host*:*port*/lmz/webservice
PollDelay	The polling delay for reverse polling (in seconds).	10

Option	Description	Example Value
AGENT_PORT	The port on which the agent listens for user provisioning requests from Select Identity.	5601
MAPPING_FILE	The XML mapping file.	Mapping.xml
SPML_DELAY	The delay (in milliseconds) between successive SPML requests sent from the agent. Increase this delay if the network or Select Identity server is performing slowly.	10000

To edit any of these values after installation, you can edit the `properties.ini` file, which resides in *`install_dir`*`\conf`.

After specifying these values, click **Next**.

10 To enable reverse synchronization, you must install the reverse notification tables. (See Operations Supported by the Connector on page 8 for an explanation of reverse synchronization.) Select the **Install reverse notification tables Now** option to install the tables. Then, click **Next** and proceed to the next step.



If you choose not to install the reverse notification tables, skip to Step 14 on page 18. (You can manually install the tables later, if necessary. This is described in Installing the Reverse Notification Tables on page 21.)

11 If you selected the **Install reverse notification tables now** option on the Reverse Table Install dialog, specify authentication information for the database user. Then, click **Next**. The tables are installed for the schema specified by the mapping file.

12 Review the installation summary for the tables. If you wish to make changes, or if the table installation failed, click **Previous** and edit the chosen options, such as the credentials. You can also select the **Show Logs** option to review the table installation log files. Then, click **Next**.

13 If you selected the **Show Logs** option, the Detailed Logs dialog is displayed. Review the log entries and click **Next**.

14 When the installation wizard completes, click **Done** on the Install Complete dialog to close the installation program.

If you configured the agent for reverse synchronization (by installing the reverse notification tables), you must edit the *install_dir*`/conf/opattributes.properties` file to configure the reverse notification tables. See Step 4 on page 21 for details.

Manual Installation

Instead of using the installation wizard, you can install the agent files and reverse notification tables manually. The following sections describe how to do this.

Installing the Agent

Complete the following steps to manually copy the agent files to the target server:

1. *On Windows:*
   Extract the contents of the `SQL-Admin-Agent-Win.zip` file, which resides in the `Manual Agent` subdirectory on the CD, to a target location for the agent on the SQL Server system. The extracted files will reside in the `SQL-Admin-Agent-Win` directory.
2. Copy the mapping file created in Step 8 on page 11 to the *`agent_home`*`/conf/com/trulogica/truaccess/connector/schema/spml` directory.
3. Modify the `properties.ini` file, which resides in the *`agent_home`*`/conf` subdirectory, to specify parameters for the agent. The parameters are listed in the following table.

Parameter	Sample Values	Description
PORT	1433	The port on which the database server is listening.
DB_DRIVER	com.microsoft.jdbc.sqlserver.SQLServerDriver	JDBC driver for the database connection.
DB_URL	jdbc:microsoft:sqlserver	JDBC URL string used for the database communication.
SERVICE	SI_DB	Database name.
CHECK_LOGIN	true	The Login Check flag.

Parameter	Sample Values	Description
MAX_LOGIN_RETRIES	3	The number of times the agent will attempt to log in to the database.
CONCERO_SERVER_URL	http://*host*:*port*/lmz/webservice	URL of the Select Identity Web Service.
PollDelay	10	The polling delay for reverse polling (in seconds).
AGENT_PORT	5601	The port on which the agent listens for user provisioning requests from Select Identity.
MAPPING_FILE	Mapping.xml	The XML mapping file.
SPML_Delay	10000	The delay (in milliseconds) between successive SPML requests sent from the agent. Increase this delay if the network or Select Identity server is performing slowly.

4 Copy the SQL Server JDBC driver files (`msbase.jar`, `msutil.jar`, and `mssqlserver.jar`) to the system CLASSPATH. Obtain these files from the SQL Server system, the Select Identity server, or your system or database administrator.

See Installing the Reverse Notification Tables on page 21 for steps to configure reverse synchronization. See Starting the Agent on page 24 for information about starting the agent.

Installing the Reverse Notification Tables

Perform these steps if you want to synchronize changes made to users in SQL Server with Select Identify. Reverse synchronization relies on reverse notification tables configured on the database. When you start the agent, reverse synchronization is enabled.

1. Copy the XSL mapping file created in Step 9 on page 11 to the *agent_home*/conf/com/trulogica/truaccess/connector/schema/spml directory.
2. Edit the properties.ini file, which resides in the *agent_home*/conf subdirectory, to specify parameters for reverse synchronization. See Step 3 on page 19 for details on this file.
3. Run the *agent_home*/Adminsetup.cmd file (on Windows) or Adminsetup.sh file (on UNIX) from the command line. This installs reverse notification tables as specified by the mapping file and creates snapshot tables. If the tables exist, table creation fails, indicating the error.
4. Modify the opattributes.properties file, which resides in the *agent_home*/conf/ subdirectory and provides operational attributes that are sent to the Select Identity server during reverse synchronization requests. The file must contain the following:

Parameter	Sample Values	Description
urn:oasis:names:tc:SPML:1:0# UserIDAndOrDomainName	Sisa	User ID of the administrative user on Select Identity
urn:trulogica:concero:2.0#password	Abc123	Password of the administrative user
urn:trulogica:concero:2.0# reverseSync	true	Set to **true** if you want to enable reverse synchronization

Parameter	Sample Values	Description
urn:trulogica:concero:2.0# resourceType	GenSQL_ Schema1	The name of the XSL file (without the .xsl extension) that is used during reverse synchronization
urn:trulogica:concero:2.0# resourceId	SQL	The name of the Select Identity resource that is created for the SQL Server Admin connector.

If you wish to delete the reverse notification tables, complete the steps in Uninstalling the Agent on page 39. These steps assume that *agent_home*/conf/properties.ini is configured as mentioned in Installing the Agent on the Database Server on page 11.

Installed Files

The following provides a listing of the directories and files installed for the agent:

Directories and Files	Description
agent_home/	Contains the following files: • `AddToStartupGroup.cmd/sh` — Adds icons to startup group; this file is present only if the agent was installed using the wizard • `CopyFile.cmd/sh` — Used by agent to copy files; this file is present only if the agent was installed using the wizard • `DelFile.cmd/sh` — Used by agent to delete files; this file is present only if the agent was installed using the wizard • `setup.cmd/sh` — Installs the reverse notification tables • `sqlapp.cmd/sh` — Used by agent to communicate with the database • `SQLConnectorConsole.cmd/sh` — Starts the agent • `AdminUninstall.cmd/sh` — Uninstalls the reverse notification tables
agent_home/conf/	Contains the following files: • `properties.ini` — Provides configuration settings for the agent • `opAttributes.properties` — Provides configuration settings for reverse synchronization • `log4j.properties` — Provides settings for logging.
agent_home/conf/com/	Contains the `trulogica/truaccess/connectior/schema/spml` directory structure where the XML mapping file is stored

Directories and Files	Description
agent_home/lib/	Contains JAR files used by the agent.
agent_home/logs	Contains log files produced by the agent.
agent_home/Uninstall_ADMIN_SQL_CONN_AGENT/	Contains files for uninstalling the agent. This subdirectory is created only if the agent is installed using the installation wizard.

Starting the Agent

To start the agent, run `SQLConnectorConsole.cmd` (on Windows) or `SQLConnectorConsole.sh` (on UNIX), which resides in the agent's home directory. This program logs in to the database server using the user name and password of a user who has administrative privileges on the database.

If you wish, you can provide the following parameters to the command:

`username` — The user name of the user who has administrative privileges on the database.

`password` — The specified user's password.

Here is an example you can use on Windows:

```
agent_home/SQLConnectorConsole.cmd -userName si -password abc123
```

If you start the agent before or without configuring reverse synchronization (the reverse notification tables), a message is displayed stating that reverse notification is disabled.

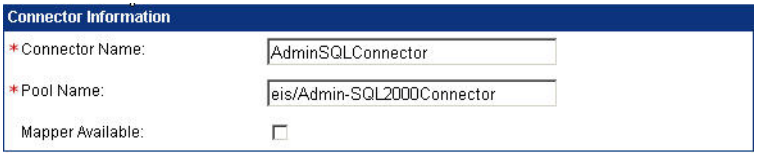
2

Configuring the Connector

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

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

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



2 Deploy a resource that uses the newly created connector. On the Resources home page, click the **Deploy New Resource** button. The resource configuration depends on how the 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.

Complete the steps in this procedure as described in the "Resources" chapter of the *HP OpenView Select Identity Administrator Guide*.When configuring the resource, refer to the following table for parameters specific to this connector:

Copy or move the XML and XSL files to the proper locations. For example, if `C:\si3.3\weblogic\sysarchive` is a folder in the WebLogic CLASSPATH, the XSL should reside in `C:\si3.3\weblogic\sysarchive` and the XML should reside in `C:\si3.3\weblogic\sysarchive\com\trulogica\truaccess\connector\schema\spml`.

Field Name	Sample Values	Description
Resource Name	Gen-SQL2000	The name of the resource.
Resource Type	SQL	The connector that was deployed in Step 1 on page 25.

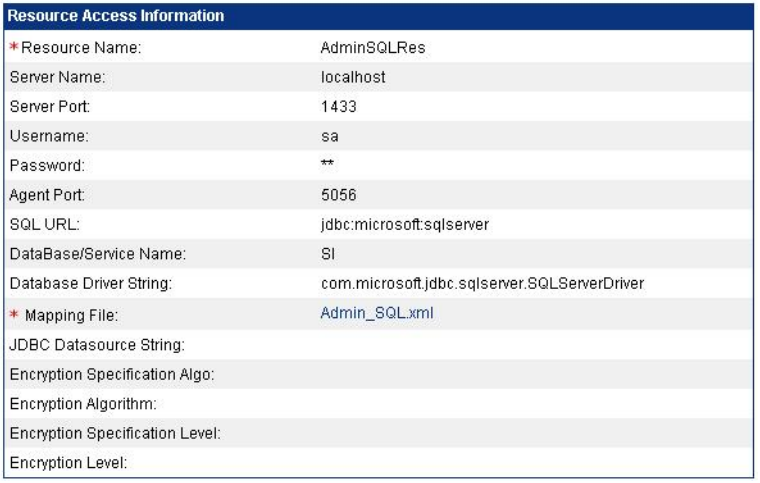
Field Name	Sample Values	Description
Authoritative Source*	No	Whether this resource is a system that is considered to be the authoritative source for user data in your environment. Specify **Yes** if the connector is enabled for reverse synchronization. If the resource is not authoritative, the resource can only modify user entitlements during reverse synchronization.
Associate to Group	Selected	Whether the system uses the concept of groups. For this connector, select this option.
Server Name	Ps0111	Host name or IP address of the database server. You must specify this parameter if the agent was installed.
Server Port	1433	Port on which the database server is listening. You must specify this parameter if the agent was installed.
Username	sa	The login name of the database administrative user. You must specify this parameter if the agent was installed.
Password	p4ssword	Password of the database administrative user. You must specify this parameter if the agent was installed.
Agent Port	5601	The port where the agent listens for incoming connections. You must specify this parameter if the agent was installed.

Field Name	Sample Values	Description
SQL URL	jdbc:microsoft:sqlserver	URL to use to communicate with the database over a JDBC connection. You must specify this parameter if the agent was installed.
Database / Service Name	testDB	The database name in which to provision users. You must specify this parameter if the agent was installed.
Database Driver String	com.microsoft.jdbc.sqlserver.SQLServerDriver	Name of the JDBC driver to connect to the database. You must specify this parameter if the agent was installed.
Mapping File	Mapping.xml	The XML mapping file, which must reside in `install/conf/com/trulogica/truaccess/connector/schema/spml` directory in order for the Select Identity server to find it.
JDBC Datasource String	Jdbc/SQLDataSource	JNDI data source name that was created or identified on the Select Identity server that can connect to the target SQL Server database. Specify a value for this property if the agent was not installed.
Encryption Specification Algo		Encryption algorithm specification string. Note that secure JDBC is not supported with this connector (you do not have to specify these parameters).
Encryption Algorithm		Name of the encryption algorithm. Note that secure JDBC is not supported with this connector (you do not have to specify these parameters).

Field Name	Sample Values	Description
Encryption Specification Level		Encryption level specification string. Note that secure JDBC is not supported with this connector (you do not have to specify these parameters).
Encryption Level		Encryption level. Note that secure JDBC is not supported with this connector (you do not have to specify these parameters).

* Instead of creating an authoritative resource, you can create authoritative attributes (in the next step) for the attributes that will be synchronized. Entitlements are authoritative by default in a non-authoritative resource but other attributes are not.

After you deploy the resource for the SQL Server Admin connector, the Access Info page of the resource properties will look similar to this:



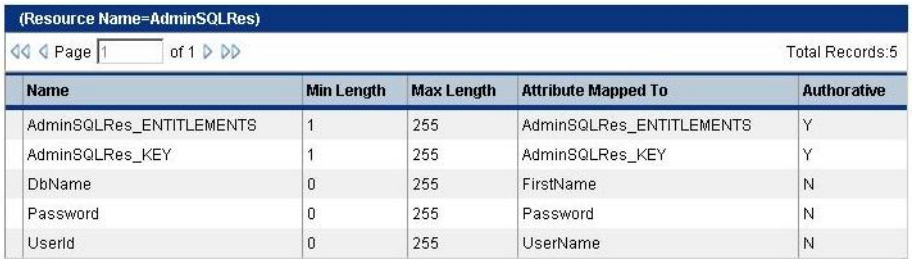
Resource Access Information	
* Resource Name:	AdminSQLRes
Server Name:	localhost
Server Port:	1433
Username:	sa
Password:	**
Agent Port:	5056
SQL URL:	jdbc:microsoft:sqlserver
DataBase/Service Name:	SI
Database Driver String:	com.microsoft.jdbc.sqlserver.SQLServerDriver
* Mapping File:	Admin_SQL.xml
JDBC Datasource String:	
Encryption Specification Algo:	
Encryption Algorithm:	
Encryption Specification Level:	
Encryption Level:	

3 Create attributes that link Select Identity to the connector. For each mapping in the connector's mapping file, create an attribute using the Attributes capability on the Select Identity client.

Refer to the "Attributes" chapter in the *HP OpenView Select Identity Administrator Guide* for more information. After you create the attributes

for the SQL Server Admin connector, the View Attributes page for the resource will look similar to this:

The attributes in the snapshot are sample values based on the sample XML file given above.



(Resource Name=AdminSQLRes)

Page 1 of 1 Total Records:5

Name	Min Length	Max Length	Attribute Mapped To	Authorative
AdminSQLRes_ENTITLEMENTS	1	255	AdminSQLRes_ENTITLEMENTS	Y
AdminSQLRes_KEY	1	255	AdminSQLRes_KEY	Y
DbName	0	255	FirstName	N
Password	0	255	Password	N
UserId	0	255	UserName	N

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

If you are enabling reverse synchronization, configure the Service as follows:

— When selecting the Business Relationship, choose the ReconciliationDefaultProcess workflow for the RECONCILIATION:Add Service and RECONCILIATION:Delete Service Membership request events. For RECONCILIATION:Add Service, use the user addition view.

— In the user addition view, specify mandatory attributes that are guaranteed to be passed by the reverse synchronization request when adding a user. If you specify a mandatory attribute that is not passed by the resource, the user will be created in Select Identity but reverse synchronization will not succeed.

— When specifying the context, obtain the value from the add request issued by the resource. For example, if the context is Country and the value is US, the <addRequest> element in the reverse synchroniation request should have an attribute called country and a value of US. If the context attribute is not present in the add user request, the user will be created in Select Identity but will not be assigned to a Service.

3

Understanding the Mapping Files

To enable the connector to provision users and entitlements in the schema on the SQL Server resource, you must create an XML mapping file. If you configured the agent to support reverse synchronization, you must also provide an XSL file that provides a reverse mapping of the Select Identity and resource fields mapped in the XML file.

This chapter provides an explanation of the XML and XSL mapping files. The following sections are provided:

Refer to `Admin_SQL.xml` and `Admin_SQL.xsl`, which were extracted from the `Admin SQL Schema.zip` file, for a sample XML and XSL files for this connector.

Elements in the XML Mapping File

Here is an explanation of the format of the XML mapping file. For a sample mapping file, see the `Admin_SQL.xml` that was extracted from the `Admin SQL Schema.zip` file.

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

  Provides standard elements for header information.

- **<objectClassDefinition>**

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

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

    — Create (CREATE)

    — Read (READ)

    — Update (UPDATE)

    — Delete (DELETE)

    — Enable (ENABLE)

    — Disable (DISABLE)

    — Reset password (RESET_PASSWORD)

    — Expire password (EXPIRE_PASSWORD)

    — Change password (CHANGE_PASSWORD)

    — Assign entitlements (LINK)

    — Unassign entitlements (UNLINK)

    — Retrieve entitlements (GETALL)

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

— true — the operation is supported by the connector

— false — the operation is not supported by the connector

— bypass — the operation is not supported by the connector

Here is an example:

```
<objectClassDefinition description="" name="User">
   <properties>
    <attr name="GETCHILDREN">
      <value>true</value>
    </attr>
    <attr name="DELETE">
      <value>true</value>
    </attr>
    <attr name="EXPIREPASSWORD">
      <value>true</value>
    </attr>
    <attr name="GETALL">
      <value>true</value>
    </attr>
...
```

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

  Each <attributeDefinitionReference> element contains the following attributes:

  — Name — the name of the reference.

  — Required— if this attribute is required in the provisioning (set to true or false).

  — Concero:tafield — the name of the Select Identity resource attribute.

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

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

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

Here is an example:

```
<memberAttributes>
  <attributeDefinitionReference concero:isKey="true"
  concero:resfield="adminproperty=USER,attribute=NAME"
  concero:tafield="UserId" encrypt="false"
  encryptionAlgorithm="" fk="" iTK="true" isPassword="false"
  name="adminpropertyUSERattributeNAME" required="true"
  supportedOperations="UNLINK,LINK,GETATTRIBUTES,
  GETPARENT,GETCHILDREN,GETALL,RESETPASSWORD,
  CHANGEPASSWORD,EXPIREPASSWORD,DISABLE,ENABLE,CREATE,
  DELETE,UPDATE" type="java.lang.String"/>
...
```

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

— The connector attribute names are specified in tafield. The value of attribute xyz is taken from the UserModel during provisioning.

— Composite attributes can be specified in the SQL Server Admin connector mapping file. To do this, specify [attr1] xxxx [attr2] as the connector attribute. This specifies that the value of the attr1 and attr2 attributes should be combined with the string xxxx to form a mapping for the specified resource field. The SQL Server Admin connector has code to handle these composite mappings.

You must specify static text (strings) in composite attributes withn brackets ({ }). Also, if no string separates two connector attributes, you must add a space that is within brackets, like this: attr1{ }attr2.

- **<attributeDefinition>**

  Defines the properties of each object's attribute. For example, the attribute definition for the Directory attribute defines that it must be between one and 50 characters in length and can contain the following letters, numbers, and characters: a-z, A-Z, 0-9, @, +, and a space.

  Here is an example:

```
<attributeDefinition
  description="adminpropertyENTITLEMENTattributeNAME"
  name="adminpropertyENTITLEMENTattributeNAME"
  type="java.lang.String">
  <properties>
    <attr name="minLength">
      <value>0</value>
    </attr>
    <attr name="maxLength">
      <value>255</value>
    </attr>
    <attr name="defaultValue">
      <value/>
    </attr>
    <attr name="pattern">
      <value><![CDATA[[a-zA-Z0-9@]+]]></value>
    </attr>
  </properties>
</attributeDefinition>
```

- **<concero:entitlementMappingDefinition>**

  Defines how entitlements are mapped to users.

- **<concero:objectStatus>**

  Defines how to assign status to a user.

- **<concero:relationshipDefinition>**

  Defines how to create relationships between users.

Refer to `Admin_SQL.xml`, which was extracted from the `Admin SQL Schema.zip` file, for a sample XML file for this connector.

Elements in the XSL Reverse Mapping File

If the agent is installed on the resource and you wish to enable reverse synchronization, you must create an XSL file to map all attributes that are specified in the XML mapping file. See the `Admin_SQL.xsl` file that was extracted from the `Admin SQL Schema.zip` file for a full sample.

Note that the elements in the XSL file are case sensitive.

You must define the user's ID field on the resource and in Select Identity. In the following example, RES_USERID is the user ID resource attribute for the user on the resource. The RES_PASSWORD is the corresponding password attribute on the resource. The following provides an example for setting these attributes:

```
<xsl:variable name="RES_USERID"
select="'adminproperty=USER,attribute=USERNAME'"/>

<xsl:variable name="RES_PASSWORD"
select="'adminproperty=USER,attribute=PASSWORD'"/>
```

SI_USERID is the Select Identity attribute for the user ID, and SI_PASSWORD is the Select Identity attribute for the password. The following shows how to set these attributes:

```
<xsl:variable name="SI_USERID" select="'USERNAME'"/>

<xsl:variable name="SI_PASSWORD" select="'PASSWORD'"/>
```

For each resource attribute, you must define a corresponding Select Identity attribute, which defines the attribute in Select Identity to which the resource attribute is mapped. The following example defines the RES_ATTR0 resource attribute and the SI_ATTR0 attribute in Select Identity:

```
<xsl:variable name="RES_ATTR0" select="'xxxxxxxxxx'"/>

<xsl:variable name="SI_ATTR0" select="'xxxxxxxxxx'"/>
```

Then, define the resource attribute, such as in this example for RES_ATTR0:

```
<xsl:when test="$ATTRNAME = $RES_ATTR0">
  <xsl:call-template name="AttributeBuilder">
    <xsl:with-param name="DSMLELEMENT" select="$DSMLELEMENT"/>
    <xsl:with-param name="ATTRNAME" select="$SI_ATTR0"/>
    <xsl:with-param name="ATTRVALUE" select="$ATTRVALUE"/>
```

```
    <xsl:with-param name="MODIFYFLAG" select="$MODIFYFLAG"/>
  </xsl:call-template>
</xsl:when>
```

Refer to the `Admin_SQL.xsl` file, which was extracted from the `Admin SQL Schema.zip` file, for a sample XSL file for this connector.

4

Uninstalling the Connector

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

- All resource dependencies are removed.
- The connector is deleted through the Connectors home page on the Select Identity client.

Uninstalling the Connector from WebLogic

Perform the following to delete a connector:

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

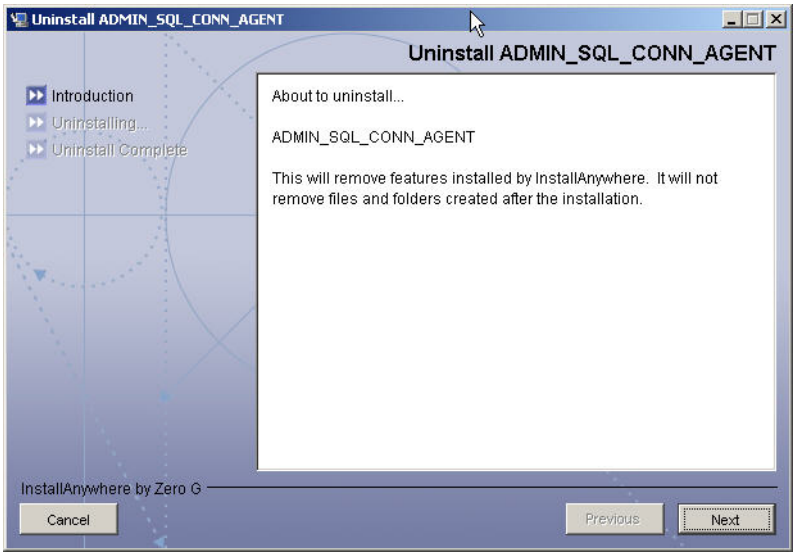
Uninstalling the Agent

The following sections describe how to remove the agent, which you can do using a wizard or manually.

Using a Wizard to Remove the Agent on Windows

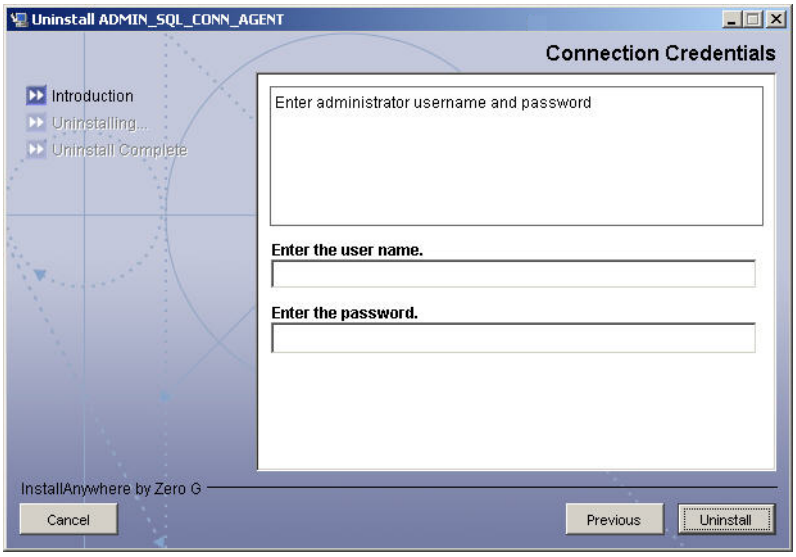
Perform the following steps to delete the agent on the Windows server:

1. Select **Programs** → **ADMIN_SQL_CONN_AGENT** → **Uninstall Agent** from the Start menu. The wizard displays.



2. Click **Next** on the introductory dialog.

3 Provide the database credentials to uninstall the reverse notification tables, if they were installed. Then, click **Uninstall**.



4 Click **Continue** when the pop-up dialog indicates that the reverse notification tables were successfully uninstalled.

5 Click **Done** on the Uninstall Complete dialog to close the wizard.

Manually Removing the Agent

Perform the following steps to manually remove the agent:

1 Make sure that the *agent_home*\conf\properties.ini file retains the same values used during the installation of the reverse notification tables.

2 Make sure that the XML mapping file during the installation of the agent is available in the *agent_home*\conf\com\trulogica\truacess\connector\schema\spml folder.

3 Run the Adminuninstall.cmd file.

4 Provide the database login credentials when prompted.

5 Delete the agent files and directory structure, if you wish.

A

Troubleshooting

This appendix describes common problems encountered during the installation and use of the connector and its agent.

Connector Installation

This section lists the common problems encountered during installation and use of the connector.

- After redeploying the connector, Select Identity does not display the current connector information.

  Possible Cause: The application is using a cached connector file.

  Solution: Restart the application server.

- Select Identity does not display the most current mapping file information.

  Possible Cause: The application server is using a cached mapping file.

  Solution: Restart the application server.

- The mapping file of a existing resource is changed and, when you attempt to modify the resource to add a new mapping file, the following error displays:

  ```
  Application cannot be modified at this time
  ```

Possible Cause: Major differences may exist between the old and new mapping files.

Solutions:

— Create a new resource with the new mapping file.

— Unmap all attributes in the current resource and modify the resource to reference the new mapping file. You cannot use this second solution, however, if users were provisioned using this resource.

- Select Identity can successfully add a user but the new user is not shown in the resource's database table.

  Possible Causes:

  — The mapping file lacks the Create operation for the Key attribute.

  — The Create operation for the User entity is not added in the XML file.

  — The XML parser files may be missing from the `BEA_HOME/jdk_1.4.1/jre/lib/endorsed` folder (on WebLogic).

  — A database exception occurred.

  Solutions:

  — Add the create operation to the mapping file or add the relevant JARs to the path.

  — If a database exception occurred, refer to the logs for details of the exception. Common exceptions include size mismatches for columns and foreign key constraint violations. Refer to the database documentation for more information on the database exceptions.

Agent and Reverse Notification Tables Installation

This section lists the common problems encountered while installing and configuring reverse synchronization.

- A NullPointerException occurs

  Possible Cause: The specified mapping file is not available in the class path.

  Solution: Make sure that the file is placed in the `Install/conf` directory. Ensure the name of the file specified in `properties.ini` is spelled correctly. Note that it is case sensitive. Also, check the format of the mapping file.

- The following error message is displayed:

  ```
  FATAL
  [com.trulogica.sql.connagents.AdminMasterScriptExecutor]: Can't
  create  view. Message received from the database:
  [Microsoft][SQLServer 2000 Driver for JDBC][SQLServer]There is
  already an object named 'DBA_USERS' in the database. Cannot
  continue.
  ```

  Possible Cause: The database user has insufficient database permissions. The reverse notification tables are already installed.

  Solution: Remove the previously deployed reverse notification tables and try again.

Agent Execution

This section lists the common problems encountered while running the agent.

- An exception similar to the following is displayed:

  ```
  java.net.BindException: Address in use: JVM_Bind
  ```

  Possible Cause: The listening port on the agent's system is in use, possibly by another invocation of the agent.

  Solution: Stop the older invocation and run the agent again.

- An error message similar to the following is displayed:

  `Invalid Object` *`schema.tableName`*

  Possible Cause: The schema specified in the mapping file is incorrect.

  Solution: Check the mapping file.

- The agent console shows a Log4jFactory exception when started.

  Possible Cause: The agent cannot find the `log4j-1.2.8.jar` in the classpath.

  Solution: Add the JAR to `BEA_HOME/jdk1_4_05/jre/lib/ext` (on WebLogic).

- The following error is displayed:

  ```
  SQLException occurred while adding element into SNAPSHOT_TAB.
  Message received from the database: ORA-00942: table or view
  does not exist
  ```

Possible Cause: The agent is installed without the reverse notification tables.

Solution: Install the tables by re-running the installation, then run the agent.