HP OpenView Select Identity Connector

Connector Version: 3.8

Installation and Configuration Guide for the Connector for Solaris™ with Telnet

hp

i n v e n t

Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notices

© Copyright 2006 Hewlett-Packard Development Company, L.P.

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

Select Identity uses software from the Apache Jakarta Project including:

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

Additional third party software used by Select Identity includes:

- JasperReports developed by SourceForge
- iText (for JasperReports) developed by SourceForge
- BeanShell
- Xalan from the Apache XML Project
- Xerces from the Apache XML Project
- Java API for XML Processing from the Apache XML Project
- SOAP developed by the Apache Software Foundation
- JavaMail from SUN Reference Implementation
- Java Secure Socket Extension (JSSE) from SUN Reference Implementation
- Java Cryptography Extension (JCE) from SUN Reference Implementation
- JavaBeans Activation Framework (JAF) from SUN Reference Implementation

- OpenSPML Toolkit from OpenSPML.org

- JGraph developed by JGraph

- Hibernate from Hibernate.org

- BouncyCastle engine for keystore management, bouncycastle.org

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

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

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

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

Trademark Notices

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

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

JAVA™ is a US trademark of Sun Microsystems, Inc.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

Oracle® is a registered US trademark of Oracle Corporation, Redwood City, California

UNIX® is a registered trademark of The Open Group.

Support

Please visit the HP OpenView support web site at:

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

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

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

- Search for knowledge documents of interest

- Submit enhancement requests online

- Download software patches

- Submit and track progress on support cases

- Manage a support contract

- Look up HP support contacts

- Review information about available services

- Enter discussions with other software customers

- Research and register for software training

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

To find more information about access levels, go to:

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

To register for an HP Passport ID, go to:

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

Contents

1 Installing the Connector

The Solaris connector enables HP OpenView Select Identity to perform the following tasks on Solaris servers running Telnet:

- 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

⚠ Do not use the user's primary group as an entitlement. If the primary group is changed, it remains as an entitlement in Select Identity but not on the resource.

The Solaris connector is a one-way connector. It pushes the changes made to user data in the Select Identity database to a target server. The mapping file maps Select Identity fields to Solaris fields.

This connector is i18n compliant provided /etc/passwd and /etc/group are in ASCII.

The Solaris connector is packaged with the following files

- UnixTelnetConnector.rar — contains the connector binary files.
- unixtelschema.jar — contains the mapping files.
- expect.solaris.linux.telnet.scripts.zip — contains the scripts used to communicate with the Solaris server.

These are located in the Solaris - Telnet directory on the Select Identity Connector CD.

System Requirements

The Solaris 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
	WebLogic 8.1.2 on Solaris 9	Oracle 9i
	WebLogic 8.1.2 on HP-UX 11i	Oracle 9i
	WebSphere 5.1.1 on Solaris 9	DB2 8.2 (or DB2 8.1 Service Pack 7)
3.3	WebLogic 8.1.4 on Windows 2003	SQL Server 2000
	WebLogic 8.1.4 on Solaris 9	Oracle 9i
3.3.1	WebLogic 8.1.4 on Windows 2003	SQL Server 2000
	WebSphere 5.1.1 on HP-UX 11i	Oracle 9i
	WebLogic 8.1.4 on Red Hat Linux AS 3.0	Oracle 10g
4.0	The Solaris connector is supported on all the platform configurations of Select Identity 4.0.	

This connector is supported on Solaris 8 and Solaris 9.

Deploying on the Application Server

To install the Solaris connector on Select Identity, you must deploy the connector on the web application server. To deploy the connector on a web application server, complete the following tasks:

1   Create a subdirectory in the Select Identity home directory where you can store the connector's Resource Adapter Archive (.rar) file.
    For example, you can create `<OVSI_HOME_DIR>`/connectors
    where `<OVSI_HOME_DIR>` = /opt/Select_Identity in Linux
    and `<OVSI_HOME_DIR>` = C:\Select_Identity in Windows (A connector subdirectory may already exist.)

2   Copy the `UnixTelnetConnector.rar` file from the Select Identity Connector CD to the connector subdirectory

3   Perform the following steps to deploy the connector on WebLogic. If deploying on WebSphere, skip to step 4 on page 9.

    a   Create a schema subdirectory in the Select Identity home directory where you can store the connector's mapping files.
        For example, you can create <OVSI_HOME_DIR>/Schema
        where <OVSI_HOME_DIR> = /opt/Select_Identity in Linux
        and <OVSI_HOME_DIR> = C:\Select_Identity in Windows (A schema subdirectory may already exist.)

b   Extract the contents of the `unixtelschema.jar` file (on the Select Identity Connector CD) to the schema subdirectory.

c   Ensure that the `CLASSPATH` environment variable in the WebLogic server startup script refers to the schema subdirectory.

d   Start the application server if it is not currently running, and log on to the WebLogic Server Console.

e   In the left pane, expand Deployments folder, and then right click on Connector Modules, and select **Deploy a New Connector Module**.

Alternatively, at the right-hand panel of the Server Console homepage, click on **Connector Modules** link, which is under Your Deployed Resources column of Domain Configurations section. Resource Connectors page appears. Click on **Deploy a New Connector Module** link on this page.

f   Click the link in the Location field, locate, and select the `UnixTelnetConnector.rar` file from the list. It is stored in the connector subdirectory.

g   Click **Target Module**.

h   If only one server is configured, skip to next step. If more than one server is configured, the next page prompts you to select the servers on which you want to deploy the connector. Select the server instance (for instance, My Server), and then click **Continue**.

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

4   If you want to deploy the connector on WebSphere, perform the following steps:

a   Stop the application server.

b   Extract the contents of the `unixtelschema.jar` file (on the Select Identity Connector CD) to `WebSphere\AppServer\lib\ext`, and then start the application server.

c   Start the application server.

d   Log on to the WebSphere Application Server Console.

e   Navigate to **Resources → Resource Adapters**.

f   Click **Install RAR**.

g   In the Server path field, enter the path to the `UnixTelnetConnector.rar` file. It is stored in the subdirectory created in step 1.

h   Click **Next**.

i   In the Name field, enter a name for the connector, and then click **OK**.

j   Click the **Save** link (at the top of the page).

k   On the Save to Master Configuration dialog, click **Save**.

l   Click **Resources → Resource Adapters**.

m   Click the new connector.

n   Click **J2C Connection Factories** in the Additional Properties table.

o   Click **New**.

p   In the Name field, enter the name of the factory for the connector. For the Solaris connector, enter **eis/UnixTelnetCon**.

q   Click **OK**.

r   Click the **Save** link.

s   On the Save to Master Configuraton dialog, click the **Save** button.

t   Restart WebSphere.

After installation, refer to Configuring the Connector on page 17 for information about registering and configuring the connector in Select Identity.

Installing Expect, Inetutils, and the Scripts

The Solaris connector performs operations over Telnet using a tool called Expect, which must be installed on the application server running Select Identity. Perform the following steps to install:

Install Expect and Inetutils

On Select Identity installed on UNIX:

1   Obtain Expect from **http://expect.nist.gov**

2   Install Expect (the `expect-5.43` package) on UNIX server

3

4   When deploying a UNIX system as a resource in the Select Identity interface, provide the location of the Expect executable on the Access Info page. TCL/TK (the `tcltk-8.4.9` package) and GCC are prerequisites for running Expect.
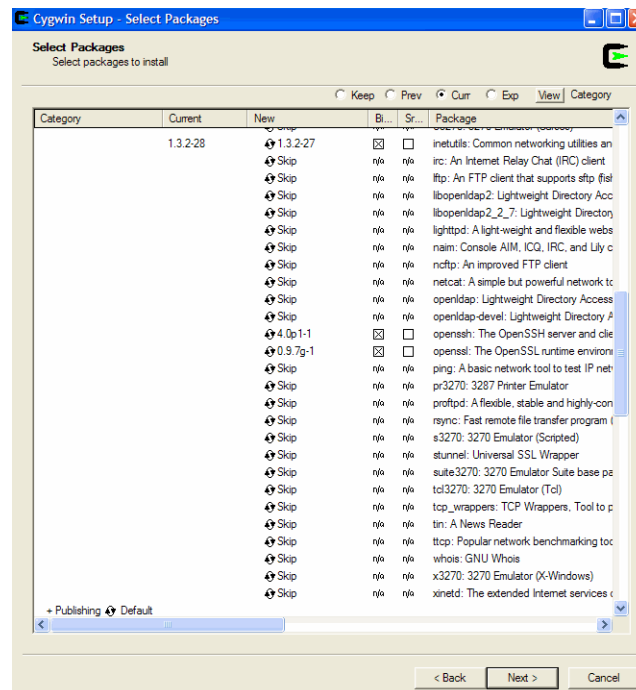
On Select Identity installed on Windows:

1   Download the Cygwin setup wizard from **http://www.cygwin.com/setup.exe**

2   Install Expect (version 5.26), which is listed in the Interpreters category in Cygwin's installer:
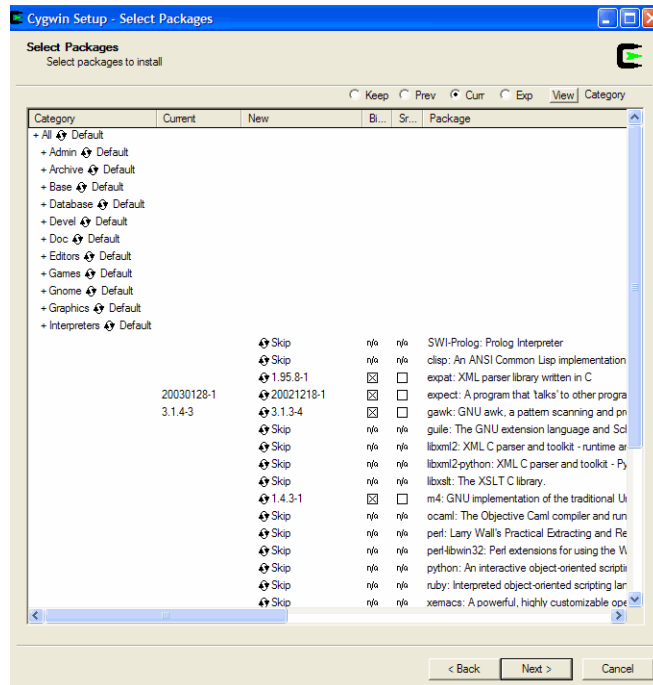
Inetutils are listed in the Net category:

▶ Ensure that Cygwin's bin directory is included in the Windows server `PATH` environment variable.

Install the scripts

Install the scripts on the application server. Copy `expect.solaris.linux.telnet.scripts.zip` from the Select Identity Connector CD and extract the scripts onto the application server.

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.

Deploying the Connector with Select Identity

Complete the following steps to deploy and configure the connector with Select Identity:

- Register a new connector with Select Identity.
- Add a resource to Select Identity.
- Link the resource attributes to the Select Identity attributes.
- Create a service that uses the newly created resource

1   To register the connector with Select Identity, perform the following steps:

    a   Click **Service Studio** → **Resources**. The Resources page appears.

    b   Click **Manage Connectors** on Resources page and add the connector. Refer to *Service Studio* chapter of *HP OpenView Select Identity Administrator Guide* for more information. While adding the connector, enter pool name as **eis/UnixTelnetCon**. Select Identity displays the connector properties in the following format:

2   To deploy a resource that uses the newly created connector, perform the following steps:

    a   Click **Service Studio** → **Resources.** Resources page appears.

    b   Click **Add New Resource**. When configuring the resource, refer to the following table for parameters specific to this connector:

Field Name	Sample Values	Description
Resource Name	Solaris8_Server	Name given to the resource.
Resource Type	Solaris8	The connector that was deployed in step 1 on page 13.

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. You must specify **No** because the connector cannot synchronize account data with the Select Identity server.
Associate to Group	Selected	Whether the system uses the concept of groups. For this connector, select this option.
Host Name	server.company.com	IP Address or host name of the Solaris machine.
User Name	accountadmin	Login account on the Solaris machine.
User Password	Password123	Password for the User Name account.
Admin Password	rootPassword	The password to gain administrator privileges.
Executable	*On UNIX:* /tools/expect-5.21/ expect.exe *On Windows:* d:/cygwin/bin/expect.exe	Path name of the Expect executable, required to run scripts.
Script Location	*On UNIX*: /connectorScripts/expect/ Solaris8/ssh/ *On Windows*: /cygdrive/c/selectidentity/ scripts/	Location of the scripts that are used by the connector. Note that on Windows, you should include a trailing slash in the path.
Mapping File	UnixConnector-tel.xml	Location of the connector mapping file used to map resource attributes to Select Identity attributes.

Refer to *Service Studio* chapter of *HP OpenView Select Identity Administrator Guide* for information on managing resource. After deploying the resource for the connector, Select Identity displays the Resource Access Information page in the following format:

3 Link the resource attributes to the Select Identity attributes.. To do this, perform the following steps.

a Click **Service Studio** → **Resources**. The Resources page appears.

b Select the newly created resource from the Resource List, and then click **Modify**. Basic Information page appears.

c Click **Resource Attribute Mapping** link on left pane. Resource Attribute Mapping page appears.

d Map each resource attribute to Select Identity Attribute using drop-down list.

Refer to the *Service Studio* chapter in *HP OpenView Select Identity Administrator Guide* for more information

> ▶ In future, if you replace the old mapping file with a new one, you need to un-map the attributes, which are not present in the new mapping file, to complete the mapping successfully.

> ⚠ You cannot un-map an attribute, if services are associated to the resource and users are added to the service using that attribute. In that case, add the attribute to the new mapping file, even though it is no longer used by the connector.

After creating the attributes for the Solaris connector, the View Attributes page for the resource looks like this:

4   To create a Service that uses the newly created resource, perform the following steps.

a   click **Service Studio** → **Services**. The Service List page appears.

b   Click **Add New Service**. Refer to *Service Studio* chapter in *HP OpenView Select Identity Administrator Guide* for more information. Reference the new resource created in step 2 while creating this service.

If the attributes in the resource do not match with the attributes defined in the xml mapping file of the connector (available in .jar file), you need to modify the mapping file as needed. See Understanding the Mapping File on page 23 for details.

Configuring Connector on Non-English Platforms

If you install the connector on non-English platform, you will have the following limitations while configuring the connector:

- When entering user attributes to provision (in the Select Identity client), you can enter local language characters except for the following attributes

  — UserName

  — Password

  — Email

- The attribute names on the resource cannot contain non-English characters. Thus, you cannot include non-English characters in the mapping file.

- Non-English entitlements are not supported by the connector.

- All configuration and property file names must be in English.

- The exception messages from the resource are in English only.

- The log messages are in English only.

- The Select Identity resource name, which is included in the reverse synchronization configuration of the agent, must be in English.

➤ Reverse synchronization of local language characters is supported. When provisioning users on the LDAP resource, you can enter local language characters as input data. These characters are reconciled with Select Identity through SPML communication. However, the following user attributes must contain English characters:

- UserName

- Password

- Email

Advanced Configuration

The Solaris connector supports passing of all mapped attributes to the find user (`findUser.exp`) and delete user (`deleteUser.exp`) expect scripts.

This advanced configuration is enabled by setting the following property in the `TruAccess.properties` file, which is present in *<OVSI_Home>*/`sysArchive` directory:

`com.hp.ovsi.connector.unix.showAttributes=true`

3 Understanding the Mapping File

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

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

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

General Information

The following operations can be performed in the mapping file:

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

Here is an explanation of the elements in the XML mapping file:

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

  Provide standard elements for header information.

- **<objectClassDefinition>**

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

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

— Create (CREATE)

— Read (READ)

— Update (UPDATE)

— Delete (DELETE)

— Enable (ENABLE)

— Disable (DISABLE)

— Reset password (RESET_PASSWORD)

— Expire password (EXPIRE_PASSWORD)

— Change password (CHANGE_PASSWORD)

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

— true — the operation is supported by the connector

— false — the operation is not supported by the connector

— bypass — the operation is not supported by the connector

Here is an example:

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

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

Each <attributeDefinitionReference> element contains the following attributes:

— name — The name of the reference.

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

— concero:tafield — The name of the attribute in Select Identity. In general, the attribute assigned to tafield should be same as the physical resource attribute, or at least the connector attribute. For example, it is recommended to have the following:

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

instead of this:

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

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

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

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

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

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

— concero:init — Set this to true if this attribute needs to be passed as part of the creation of the user. You can use this parameter to control which attributes must be specified during creation and which must be specified when a user is modified.

— concero:isPassword — Set this to true if the attribute is password.

— concero:isMulti — Set this to true if the resource attribute is muli-valued.

— concero:isSensitive — Set this to true if the attribute is case-sensitive.

Here is an example:

```
<memberAttributes>
<attributeDefinitionReference name="GroupName"
required="true" concero:tafield="GroupName"
concero:resfield="gname" concero:isKey="true" />
</memberAttributes>
```

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 Solaris connector has code to interpret the mappings in one way, as follows:

— The connector attribute names are specified in square braces, like this: [xyz]. The value of attribute xyz is taken from the SIUserModel class in Select Identity during provisioning.

— Composite attributes can be specified in the Solaris 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. Solaris connector has code to handle these composite mappings.

- **<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 excerpt from the `UnixConnector-tel.xml` file:

```
<attributeDefinition name="Directory" description="Directory"
  type="xsd:string">
  <properties>
    <attr name="minLength">
      <value>1</value>
    </attr>
    <attr name="maxLength">
      <value>50</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.

Solaris Mapping Information

The Solaris connector supports the following identify information to be provisioned on the Solaris system. You can add, modify, or delete attributes once you are familiar with the contents of this file. You can edit the Select Identity resource attributes; they reflect the identity information as seen in Select Identity. The physical resource attributes are literal attributes of user accounts on the Solaris server. These attributes cannot be changed.

➤ When adding a user in Select Identity, do not specify an entitlement value (secondary group) that is the same value as the Default Group for the system resource. This may cause an entitlement to be removed inadvertently from the user if the user is modified and the Default Group value is changed for that user.

Select Identity Resource Attribute	Connector Attribute	Attribute on Solaris Resource	Description
User Name	username	login (login argument)	UNIX logon name
Password	password	password	Logon password
First Name	F		First Name
Last Name	L		Last Name
[First Name] [Last Name]	comment	login comment (-c option)	Comment section in /etc/passwd
Directory	directory	login home directory (-d option)	User's home directory
Shell	shell	login shell (-s option)	UNIX logon shell
defaultgroup	defaultgroup	login primary group (-g option)	Default group membership

4 Understanding the Installed Scripts

The Solaris connector performs operations using a tool called Expect. This tool must be installed on the application server running Select Identity as described in Installing Expect, Inetutils, and the Scripts on page 10.

Each script is explained below with a description and example of its usage from the command line. Examples are tested and verified with the following arguments:

Expect executable location : E:/cygwin/bin/expect.exe

Scripts directory: /cygdrive/e/telnet_scripts/HPUX_SSH

UNIX system name: sihp1

UNIX user name : siuser

UNIX password :abcd

UNIX root password : abcd

You can run the examples on command line by replacing the arguments with those from your environment.

- `adduser.exp`

  This is the script used to add a new user to the Solaris system. The following arguments are required:

  directory — User's home directory

  defaultgroup — User's default group

  password — User's password

  F — User's first name

  username — User's login name on the UNIX system

  shell — User's default shell

  comment — User's comment

  L — User's last name

  Example:
  ```
  E:/cygwin/bin/expect.exe -f /cygdrive/e/telnet_scripts/HPUX_SSH/
  adduser.exp sihp1 siuser abcd abcd directory=/tmp/june0601
  defaultgroup=bin password=abcd F=FirstName username=june0601 shell=/bin/sh
  comment=FirstName LastName L=LastName
  ```

- `listuser.exp`

  This script lists all users configured on the Solaris system.

  Example:
  ```
  E:/cygwin/bin/expect.exe -f /cygdrive/e/telnet_scripts/HPUX_SSH/
  listusers.exp sihp1 siuser abcd abcd
  ```

- `changepassword.exp`

This script changes the password of a user on the Solaris system. The following arguments are required:

username — User's login name on the UNIX system

password — User's new password

Example:
```
E:/cygwin/bin/expect.exe -f /cygdrive/e/telnet_scripts/HPUX_SSH/
changepasswd.exp sihp1 siuser abcd abcd june0601 abcd
```

- `changestatus.exp`

This script changes a users status on the Solaris system. The following arguments are required:

username — User's login name on the UNIX system

status — User's enable (true) OR disable(false) status

shell — User's shell for enable and /bin/false for disable

Examples:
```
E:/cygwin/bin/expect.exe -f /cygdrive/e/telnet_scripts/HPUX_SSH/
changestatus.exp sihp1 siuser abcd abcd username=june0601 status=false
shell=/bin/false
```
```
E:/cygwin/bin/expect.exe -f /cygdrive/e/telnet_scripts/HPUX_SSH/
changestatus.exp sihp1 siuser abcd abcd username=june0601 status=true
shell=/bin/sh
```

- `deleteuser.exp`

This script deletes a user from the Solaris system. The following argument is required:

username — User's login name on the UNIX system

Example:
```
E:/cygwin/bin/expect.exe -f /cygdrive/e/telnet_scripts/HPUX_SSH/
deleteuser.exp sihp1 siuser abcd abcd username=june0601
```

- `dotest.exp`

This script is executed whenever a new connector is created. It is responsible for validating the connection.

Example:
```
E:/cygwin/bin/expect.exe -f /cygdrive/e/telnet_scripts/HPUX_SSH/dotest.exp
sihp1 siuser abcd abcd
```

- `finduser.exp`

This script is used to search for a user on the Solaris system. The following argument is required:

username — User's login name on UNIX system

Example:
```
E:/cygwin/bin/expect.exe -f /cygdrive/e/telnet_scripts/HPUX_SSH/
finduser.exp sihp1 siuser abcd abcd june0601
```

- `genericcmd.exp`

  This script is used to enable the execution of any command required by Select Identity. The following argument is required:

  command — Command that needs to be executed; if a space exists in the command, quote the command

  Example:
  ```
  E:/cygwin/bin/expect.exe -f /cygdrive/e/telnet_scripts/HPUX_SSH/
  genericcmd.exp sihp1 siuser abcd abcd "/bin/cat /etc/group"
  ```

- `modifyuser.exp`

  This script is used to modify a user on the Solaris system. The following arguments are required:

  directory — User's new home directory

  defaultgroup — User's new default group

  password — User's password

  F — User's new first name

  username — User's login name on the UNIX system

  shell — User's new default shell

  comment — User's new comment

  L — User's new last name

  gname — Group names separated by commas to which the user needs to be added

  Examples:
  ```
  E:/cygwin/bin/expect.exe -f /cygdrive/e/telnet_scripts/HPUX_SSH/
  modifyuser.exp sihp1 siuser abcd abcd directory=/home/june0601
  defaultgroup=bin password=abcd F=Ravindranatha username=june0601 shell=/
  bin/ksh "comment=Ravindranatha K" L=K
  ```

  ```
  E:/cygwin/bin/expect.exe -f /cygdrive/e/telnet_scripts/HPUX_SSH/
  modifyuser.exp sihp1 siuser abcd abcd gname=adm,daemon,mail,lp,root
  username=june0601
  ```

- `getgroups.exp`

  This script is used to get all group names for which the current user has membership on the Solaris system. The following argument is required:

  username — User's login name on the UNIX system

  Example:
  ```
  E:/cygwin/bin/expect.exe -f /cygdrive/e/telnet_scripts/HPUX_SSH/
  getgroups.exp sihp1 siuser abcd abcd june0601
  ```

5 Uninstalling the Connector

If you need to uninstall a connector from Select Identity, make sure that :

- All resource dependencies are removed.

- The connector is deleted using the Select Identity client Connectors pages.

On WebLogic

Perform the following to delete a connector on WebLogic:

1   Log on to the WebLogic Server Console.

2   Expand the Deployments folder on the left pane, and then double click on **Connector Modules**

   Alternatively, at the right panel of the Server Console homepage, click on **Connector Modules** link, which is under Your Deployed Resources column of Domain Configurations section.

3   The right hand pane of the console displays a table showing all the deployed connectors. Click the delete icon next to the connector that you want to uninstall.

4   Click **Yes** to confirm the deletion.

5   Click **Continue**.

On WebSphere

Perform the following steps to uninstall the connector on WebSphere:

1   Log on to the WebSphere Application Server Console.

2   Navigate to **Resources** → **Resource Adapters**.

3   Select the connector to uninstall.

4   Click **Delete**.

5   Click the **Save** link (at the top of the page).

6   On the Save to Master Configuration dialog, click the **Save** button.

After deleting the connector, you can remove the Expect scripts as well. Remove the scripts from the directory on the Select Identity server where they were extracted (see Installing Expect, Inetutils, and the Scripts on page 10).

A Troubleshooting

The following information is provided to help you diagnose connectivity and configuration problems that you may encounter when using the connector to provision users on the resource. You may need to consult your system or web application administrator for detailed help.

- On the Select Identity server, if installed on a Windows system, ensure that Cygwin's `bin` directory is included in the Windows server's PATH environment variable.

- If an error indicating that `org.apache.log4j.Logger` could not be found is displayed while deploying the connector, be sure to exclude `log4.jar` from the Java classpath.

- Verify that the path to Telnet is configured properly on the Solaris system. Use the **which telnet** command to find the path to Telnet. If the path is different than the path specified in the Expect scripts, use the path found on the Solaris system. You may want to specify the full path to Telnet in the scripts.

- Ensure that the Telnet daemon is configured and running on the target Solaris system. Use **ps -ef | grep telnetd** to determine if the daemon is running.

- The default UNIX prompt of resource system should be $, #, %, or >.

- Verify that Telnet allows enough connections to handle provisioning requests. The number of connections depends on the number of connections allowed by the application server for the connector. To view the connection configuration and connections on the application server, perform one of the following procedures:

  On WebLogic:

  To view the connection configuration:

  a   Log on to the WebLogic Server Console.

  b   Navigate to **My_domain** → **Services** → **JDBC** → **Connection Pools** → **connection_pool**.

  To monitor current connections:

  a   Navigate to **My_domain** → **Deployments** → **Connector Modules** → **connector**.

  b   Click the **Monitoring** tab to view connections.

  On WebSphere:

  To view the connection configuration:

  a   Log on to the WebSphere Application Server Console.

  b   Click **Resources** → **Resource Adapters**.

  c   Click the connector.

  d   Click **J2C Connection Factories** in the Additional Properties table.

  To monitor current connections, you must use the Tivoli performance viewer and click **application_server** → **J2C_connection_pool**.

Consult with your system or application server administrator for further help diagnosing problems related to connections that do not close, connections that time out, connections that cannot be opened, and so on.

- Depending on the system implementation and software versions, you may experience connectivity problems when the connector communicates with the Solaris system. If so, edit each script that includes the following:

```
set force_conservative 0 ;
```

Set the force_conservative property to **1**, as follows:

```
set force_conservative 1 ;
```

This forces Expect to pause (1/10th of second) before sending data. Some processes, such as Telnet, may ignore keystrokes if they are sent too quickly. Thus, the short delay prevents the keystroke from arriving too quickly, before Expect is prepared to receive it.