HP OpenView Select Identity

Connector for Solaris™ 9 Systems with Telnet

Installation and Configuration Guide

Connector Version: 3.4
Select Identity Version: 3.3.1

August 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

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.

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.

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

Installing the Connector

The Solaris 9 connector enables HP OpenView Select Identity to perform the following tasks on Solaris 9 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

⚠ After the connector adds a user on the Solaris 9 resource, the user's primary group should not be used as an entitlement because, if the primary group is changed, the primary group will remain as an entilement in Select Identity but not on the resource.

The Solaris 9 connector is a one-way connector and pushes changes made to user data in the Select Identity database to a target server. The mapping file controls how Select Identity fields are mapped to Solaris 9 fields.

The Solaris 9 connector is packaged in the following files:

- `unixtelschema.jar` — contains the mapping files
- `UnixTelnetConnector.rar` — contains the connector binary files
- `expect.solaris.linux.telnet.scripts.zip` — contains the scripts used to communicate with the Solaris 9 server

These files are located in the `Unix Solaris 9 - Telnet` directory on the Select Identity Connector CD.

System Requirements

The Solaris 9 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

This connector is supported with Solaris 9.

Deploying on the Web Application Server

To install the Solaris 9 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 `UnixTelnetConnector.rar` file from the Select Identity Connector CD to the connector subdirectory.
3. If deploying the connector on WebLogic, complete the following steps. If deploying on WebSphere, skip to Step 4 on page 10.
   - a. Create a schema subdirectory in the Select Identity home directory where the connector's mapping file(s) will reside. For example, you could create the `C:\Select_Identity\schema` folder. (This subdirectory may 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 references the schema subdirectory.
   - d. Start the application server if it is not currently running.
   - e. Log on to the WebLogic Server Console.
   - f. Navigate to ***My_domain*** **Deployments** **Connector Modules**.
   - g. Click **Deploy a New Connector Module**.
   - h. Locate and select the `UnixTelnetConnector.rar` file from the list. It is stored in the connector subdirectory.
   - i. Click **Target Module**.
   - j. Select the **My Server** (your server instance) check box.
   - k. Click **Continue**. Review your settings.
   - l. Keep all default settings and click **Deploy**. The Status of Last Action column should display Success.

4 If deploying the connector on WebSphere, complete 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`.
- 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.
- j Click **OK**.
- k Click the **Save** link (at the top of the page).
- l On the Save to Master Configuraton dialog, click the **Save** button.
- m Click **Resources** **Resource Adapters**.
- n Click the new connector.
- o Click **J2C Connection Factories** in the Additional Properties table.
- p Click **New**.
- q In the Name field, enter the name of the factory for the connector. For the SQL connector, enter **`eis/UnixTelnetCon`**.
- r Click **OK**.
- s Click the **Save** link.
- t On the Save to Master Configuraton dialog, click the **Save** button.
- u Restart WebSphere.

5 Modify the mapping file, if necessary. See Understanding the Mapping File on page 19 for details.

After installing the connector, see Configuring the Connector on page 14 to register and configure the connector in Select Identity.

Installing Expect, inetutils, and the Scripts

The Solaris 9 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:

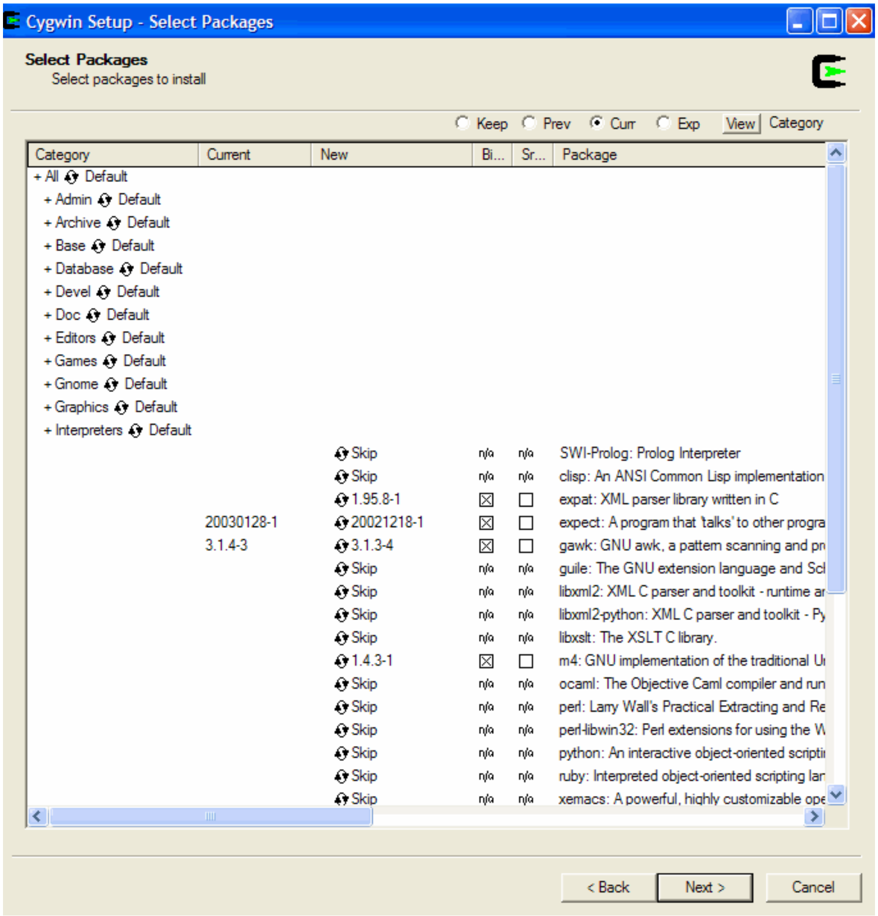
1 Install Expect and Telnet, if they were not previously installed, as follows:

On the Select Identity server installed on UNIX:
Install Expect on the UNIX server; obtain Expect from the operating system installation media. When deploying the UNIX system as a resource in the Select Identity interface, you will provide the location of the Expect executable on the Access Info page. TCL/TK and GCC are prerequisites for running Expect.

Telnet is provided by the operating system.

On the Select Identity server installed on Windows:
Download the Cygwin setup wizard from **http://www.cygwin.com/setup.exe**. Be sure to install Expect (version 5.26) and the inetutils

package. Expect is listed in the Interpreters category in Cygwin's installer:



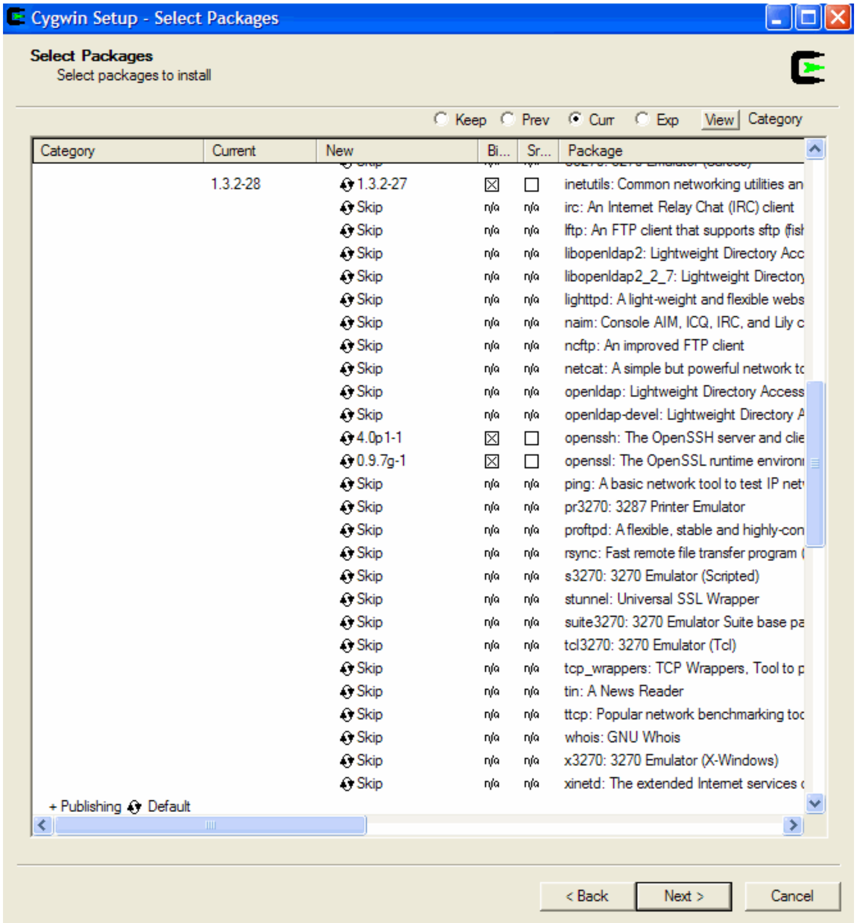
Cygwin Setup - Select Packages

Select Packages
Select packages to install

○ Keep ○ Prev ● Curr ○ Exp View Category

Category	Current	New	Bi...	Sr...	Package
+ All ⟳ Default					
+ Admin ⟳ Default					
+ Archive ⟳ Default					
+ Base ⟳ Default					
+ Database ⟳ Default					
+ Devel ⟳ Default					
+ Doc ⟳ Default					
+ Editors ⟳ Default					
+ Games ⟳ Default					
+ Gnome ⟳ Default					
+ Graphics ⟳ Default					
+ Interpreters ⟳ Default					
		⟳ Skip	n/a	n/a	SWI-Prolog: Prolog Interpreter
		⟳ Skip	n/a	n/a	clisp: An ANSI Common Lisp implementation
		⟳ 1.95.8-1	☒	□	expat: XML parser library written in C
	20030128-1	⟳ 20021218-1	☒	□	expect: A program that 'talks' to other progra
	3.1.4-3	⟳ 3.1.3-4	☒	□	gawk: GNU awk, a pattern scanning and pr
		⟳ Skip	n/a	n/a	guile: The GNU extension language and Sc
		⟳ Skip	n/a	n/a	libxml2: XML C parser and toolkit - runtime ar
		⟳ Skip	n/a	n/a	libxml2-python: XML C parser and toolkit - Py
		⟳ Skip	n/a	n/a	libxslt: The XSLT C library.
		⟳ 1.4.3-1	☒	□	m4: GNU implementation of the traditional U
		⟳ Skip	n/a	n/a	ocaml: The Objective Caml compiler and run
		⟳ Skip	n/a	n/a	perl: Larry Wall's Practical Extracting and Re
		⟳ Skip	n/a	n/a	perl-libwin32: Perl extensions for using the W
		⟳ Skip	n/a	n/a	python: An interactive object-oriented scripti
		⟳ Skip	n/a	n/a	ruby: Interpreted object-oriented scripting lar
		⟳ Skip	n/a	n/a	xemacs: A powerful, highly customizable ope

inetutils are listed in the Net category:



Cygwin Setup - Select Packages

Select Packages

Select packages to install

○ Keep ○ Prev ● Curr ○ Exp View Category

Category	Current	New	Bi...	Sr...	Package
	1.3.2-28	1.3.2-27	☒	□	inetutils: Common networking utilities an
		Skip	n/a	n/a	irc: An Internet Relay Chat (IRC) client
		Skip	n/a	n/a	lftp: An FTP client that supports sftp (fisl
		Skip	n/a	n/a	libopenldap2: Lightweight Directory Acc
		Skip	n/a	n/a	libopenldap2_2_7: Lightweight Directory
		Skip	n/a	n/a	lighttpd: A light-weight and flexible webs
		Skip	n/a	n/a	naim: Console AIM, ICQ, IRC, and Lily c
		Skip	n/a	n/a	ncftp: An improved FTP client
		Skip	n/a	n/a	netcat: A simple but powerful network tc
		Skip	n/a	n/a	openldap: Lightweight Directory Access
		Skip	n/a	n/a	openldap-devel: Lightweight Directory A
		4.0p1-1	☒	□	openssh: The OpenSSH server and clie
		0.9.7g-1	☒	□	openssl: The OpenSSL runtime environ
		Skip	n/a	n/a	ping: A basic network tool to test IP net
		Skip	n/a	n/a	pr3270: 3287 Printer Emulator
		Skip	n/a	n/a	proftpd: A flexible, stable and highly-con
		Skip	n/a	n/a	rsync: Fast remote file transfer program (
		Skip	n/a	n/a	s3270: 3270 Emulator (Scripted)
		Skip	n/a	n/a	stunnel: Universal SSL Wrapper
		Skip	n/a	n/a	suite3270: 3270 Emulator Suite base pa
		Skip	n/a	n/a	tcl3270: 3270 Emulator (Tcl)
		Skip	n/a	n/a	tcp_wrappers: TCP Wrappers, Tool to p
		Skip	n/a	n/a	tin: A News Reader
		Skip	n/a	n/a	ttcp: Popular network benchmarking toc
		Skip	n/a	n/a	whois: GNU Whois
		Skip	n/a	n/a	x3270: 3270 Emulator (X-Windows)
		Skip	n/a	n/a	xinetd: The extended Internet services (
+ Publishing Default					

2 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.

3 *On Windows:*
Ensure that Cygwin’s `bin` directory is included in the Windows server PATH environment variable. If you do not add this directory to the PATH variable, the Telnet program will not be available to the Expect script, which communicates with the UNIX resource; all provisioning activities to the resource will fail.

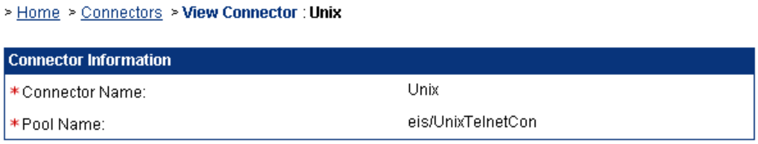
2

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

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



> Home > Connectors > **View Connector** : **Unix**

Connector Information	
*Connector Name:	Unix
*Pool Name:	eis/UnixTelnetCon

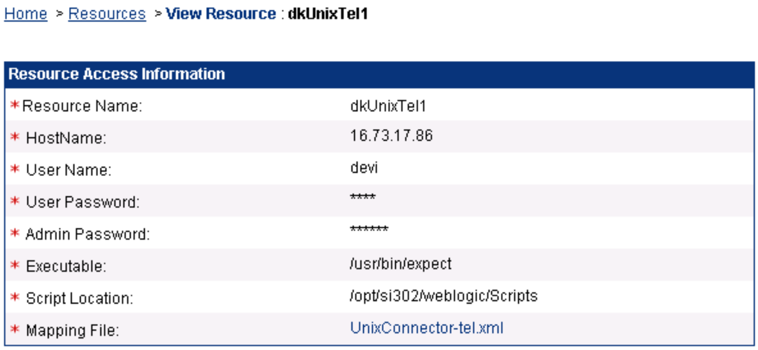
2 Deploy a resource that uses the newly created connector. On the Resources home page, click the **Deploy New Resource** button. When configuring the resource, refer to the following table for parameters specific to this connector:

Field Name	Sample Values	Description
Resource Name	Solaris9_Server	Name given to the resource.
Resource Type	Solaris9	The connector that was deployed in Step 1 on page 14.
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 9 machine.
UserName	accountadmin	Login account on the Solaris 9 machine.
UserPassword	Password123	Password for the UserName account.
AdminPassword	rootPassword	The password to gain administrator privileges.
Executable	/tools/expect-5.21/expect.exe	Path name of the Expect executable, required to run scripts.

Field Name	**Sample Values**	**Description**
scriptLocation	*On UNIX*: /connectorScripts/expect/Solaris9 *On Windows*: /cygdrive/c/selectidentity/scripts/	Location of the Expect scripts that are used by the connector. Note that on Windows, you should include a trailing slash in the path.
mappingFile	UnixConnector-tel.xml	Location of the connector mapping file used to map resource attributes to Select Identity attributes.

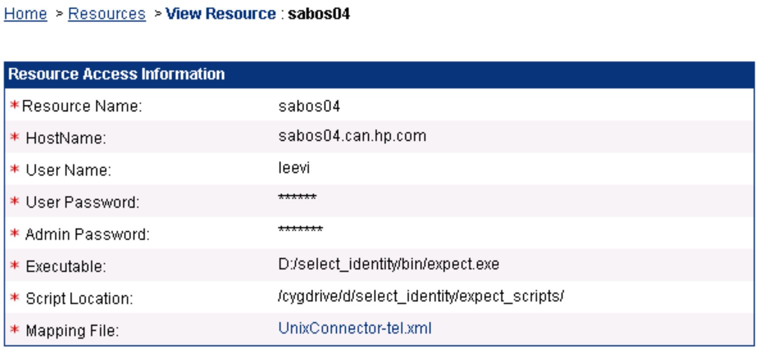
Complete the steps in this procedure as described in the “Resources” chapter of the *HP OpenView Select Identity Administrator Guide*. After you deploy the resource, the Access Info properties for the resource will look similar to the following snapshots.

On a Select Identity server installed on UNIX:



Resource Access Information	
* Resource Name:	dkUnixTel1
* HostName:	16.73.17.86
* User Name:	devi
* User Password:	****
* Admin Password:	******
* Executable:	/usr/bin/expect
* Script Location:	/opt/si302/weblogic/Scripts
* Mapping File:	UnixConnector-tel.xml

On a Select Identity server installed on Windows:



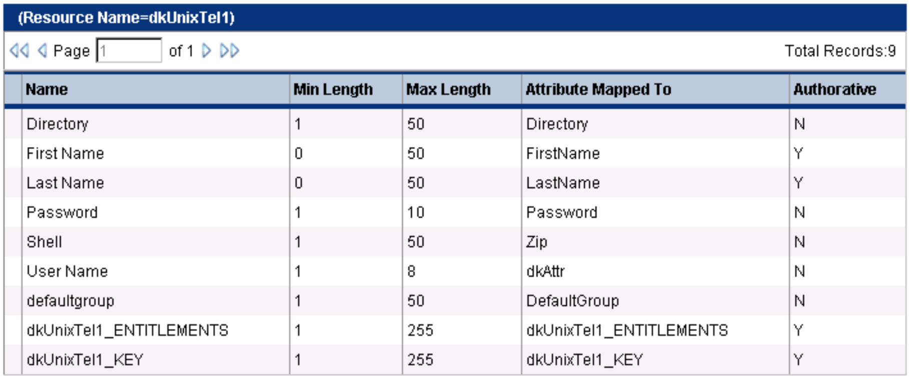
Home > Resources > View Resource : sabos04

Resource Access Information	
* Resource Name:	sabos04
* HostName:	sabos04.can.hp.com
* User Name:	leevi
* User Password:	******
* Admin Password:	*******
* Executable:	D:/select_identity/bin/expect.exe
* Script Location:	/cygdrive/d/select_identity/expect_scripts/
* Mapping File:	UnixConnector-tel.xml

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

> If a new mapping file is associated with the resource in the future (using the Modify Resource page of the Select Identity client), keep the following points in mind:
>
> - If an attribute in the old mapping file was mapped in Select Identity and that attribute is no longer included in the new mapping file, you must unmap the attribute before modifying the resource to include the new mapping file. Unmap attributes on the Modify Attribute page of the Select Identity client.
> - You cannot unmap an attribute if Services are associated with the resource and users were added to the Service. Thus, you must add the attribute to the new mapping file, even though that attribute is no longer used by the connector.

After you create the attributes for the Solaris 9 connector, the View Attributes page for the resource will look similar to this:



(Resource Name=dkUnixTel1)

◁◁ ◁ Page 1 of 1 ▷ ▷▷ Total Records:9

Name	Min Length	Max Length	Attribute Mapped To	Authorative
Directory	1	50	Directory	N
First Name	0	50	FirstName	Y
Last Name	0	50	LastName	Y
Password	1	10	Password	N
Shell	1	50	Zip	N
User Name	1	8	dkAttr	N
defaultgroup	1	50	DefaultGroup	N
dkUnixTel1_ENTITLEMENTS	1	255	dkUnixTel1_ENTITLEMENTS	Y
dkUnixTel1_KEY	1	255	dkUnixTel1_KEY	Y

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.

3

Understanding the Mapping File

The Solaris 9 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 9 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>**

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

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

instead of this:

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

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

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

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

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

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

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

Here is an example:

```
<memberAttributes>
  <attributeDefinitionReference name="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 Linux connector has code to interpret the mappings in one way, as follows:

- The connector attribute names are specified in square braces, like this: [xyz]. The value of attribute xyz is taken from the UserModel during provisioning.
- Composite attributes can be specified in the Linux 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 9 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 9 Mapping Information

The Solaris 9 connector supports the following identify information to be provisioned on the Solaris 9 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 9 server. These attributes cannot be changed. See the *HP OpenView Select Identity Connector Developer Guide* for more information about attributes and mapping information.

➤ 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 Tru64 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

Select Identity Resource Attribute	Connector Attribute	Attribute on Tru64 Resource	Description
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 9 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 11.

Each script is explained below, including 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 : iso*help

UNIX root password : iso*help

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 9 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 iso*help iso*help directory=/
tmp/june0601 defaultgroup=bin password=iso*help F=FirstName
username=june0601 shell=/bin/sh comment=FirstName LastName
L=LastName
```

- `listuser.exp`

  This script lists all users configured on the Solaris 9 system.

  Example:

  ```
  E:/cygwin/bin/expect.exe -f /cygdrive/e/telnet_scripts/
  HPUX_SSH/listusers.exp sihp1 siuser iso*help iso*help
  ```

- `changepassword.exp`

  This script changes the password of a user on the Solaris 9 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 iso*help iso*help
  june0601 iso*help
  ```

- `changestatus.exp`

  This script changes a users status on the Solaris 9 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 iso*help iso*help
username=june0601 status=false shell=/bin/false
```

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

- `deleteuser.exp`

  This script deletes a user from the Solaris 9 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 iso*help iso*help
  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 iso*help iso*help
  ```

- `finduser.exp`

  This script is used to search for a user on the Solaris 9 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 iso*help iso*help 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 iso*help iso*help "/bin/
  cat /etc/group"
  ```

- `modifyuser.exp`

  This script is used to modify a user on the Solaris 9 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 iso*help iso*help
  directory=/home/june0601 defaultgroup=bin password=iso*help
  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 iso*help iso*help
  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 9 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 iso*help iso*help june0601
  ```

5

Uninstalling the Connector

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

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

On 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**.

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 11).

On WebSphere

Complete 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 Configuraton 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 11).

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.
- Verify that the paths to Telnet is configured properly on the Solaris 9 system. Use the **`which telnet`** command to find the path of Telnet. If the path is different than the path specified in the Expect scripts, use the path found on the Solaris 9 system. You may want to specify the full path to Telnet in the scripts.
- Ensure that the Telnet daemon are is configured and running on the target Solaris 9 system. Use **`ps -ef | grep telnetd`** to determine if the daemon is running.
- Verify that Telnet allows for 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:

c Navigate to ***My_domain*** **Deployments** **Connector Modules** ***connector***.

d 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 9 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.