HP OpenView Select Identity

Connector for Red Hat Linux Systems with SSH

Installation Guide

Software Version: 3.0.1

October 2004

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

© 2004 Hewlett-Packard Development Company, L.P.

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

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

Select Identity uses software from the Apache Jakarta Project including:

- Commons-beanutils.
- Commons-collections.
- Commons-logging.
- Commons-digester.
- Commons-httpclient.

- Element Construction Set (ecs).
- Jakarta-poi.
- Jakarta-regexp.
- Logging Services (log4j).

Additional third party software used by Select Identity includes:

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

Support

Please visit the HP OpenView web site at:

http://openview.hp.com/

There you will find contact information and details about the products, services, and support that HP OpenView offers.

You can go directly to the support web site at:

http://support.openview.hp.com/

The support web site includes:

- Downloadable documentation
- Troubleshooting information
- Patches and updates
- Problem reporting
- Training information
- Support program information

contents

1

Installing the Connector

The Linux connector enables HP OpenView Select Identity to manage user data in Linux, version 8, systems. It is a one-way connector and pushes changes made to user data in the Select Identity database to a target UNIX server. The mapping file controls how Select Identity fields are mapped to Linux fields.

The Linux connector is packaged in the following files:

- `unixschema.jar` – contains the mapping files
- `UnixConnector.rar` – contains the connector binary files
- `Linux-bsh-scripts.zip` – contains the scripts used to communicate with the Linux server

These files are located in the `Linux - SSH` directory on the Select Identity Connector CD.

Deploying on the Web Application Server

To install the Linux connector on the Select Identity server, complete these steps.

➤ • Perform this procedure after the Select Identity product installation. The application server used in this procedure is WebLogic 8.1, therefore you must be familiar with the WebLogic platform.

• Make sure that SSH is installed on all Linux systems to which you want Select Identity to connect and store information.

1 A `Select_Identity` directory was created on the application server during the product installation. Create a `connector` folder in this directory.

2 Copy the `UnixConnector.rar` file to the `connector` folder in the `Select _Identity` directory.

3 Copy the `unixschema.jar` file to a temporary directory.

4 Extract the `unixschema.jar` contents to the `Select_Identity` directory.

5 Ensure that the CLASSPATH environment variable in the WebLogic server startup script references the `Select_Identity` directory.

6 Modify the mapping file, if necessary. See Understanding the Mapping File on page 10 for details.

7 Start the application server if it is not currently running.

8 Log on to the WebLogic Server Console.

9 Navigate to *My_domain* → **Deployments** → **Connector Modules**.

10 Click **Deploy a New Connector Module**.

11 Locate and select the `UnixConnector.rar` file from the list. It is stored in the `Select_Identity\connector` directory.

12 Click **Target Module**.

13 Configure the connector according to your environment. Open **Edit Connector Descriptor**, and add the following connector parameters:

— Connector Name: Specify any name for the Linux connector

— Pool Name: **eis/UnixSSH**

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

15 Click **Continue**. Review your settings.

16 Keep all default settings and click **Deploy**.

The Status of Last Action column should display Success.

After installing the connector, log on to the Select Identity client and deploy the connector using the Connector pages. Then, create a resource that represents the connector, and configure a Service that relies on the Linux resource. See the *HP OpenView Select Identity Administrator Guide* for procedures. The Resource Access Information appendix provides detailed information about creating a Linux resource.

Installing Scripts on the Web Application Server

The Linux connector uses a Java-based scripting engine called Bean Shell (**www.beanshell.org**). The Bean Shell scripts control the interactions between the Linux machine and the Select Identity server.

Copy the Linux-bsh-scripts.zip file from the Select Identity Connector CD and extract the scripts onto the application server running Select Identity.

2

Understanding the Mapping File

The Linux connector is deployed with the `UnixConnector.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 `unixschema.jar`. The mapping file is used to map user account additions and modifications from Select Identity to the system resource. When you deploy a resource using the Resources page of the Select Identity client, you can review this file.

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

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

General Information

The following operations can be performed in the mapping file:

- Add a new attribute mapping

- Delete an existing attribute mapping

- Modify attribute mappings

Here is an explanation of the elements in the XML mapping 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 and will throw a permission exception

— bypass — the operation is not supported by the connector but will not throw any exception; the operation is simply bypassed

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.

  — 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 <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 name="GroupName"
required="true" concero:tafield="GroupName"
concero:resfield="gname" concero:isKey="true" />
  </memberAttributes>
  </objectClassDefinition>
- <attributeDefinition name="GroupName"
description="GroupName" type="xsd:string">
   - <properties>
     - <attr name="minLength">
        <value>1</value>
      </attr>
     - <attr name="maxLength">
        <value>8</value>
      </attr>
     - <attr name="pattern">
       - <value>
         - <![CDATA[ [a-zA-Z0-9@]+ ]]>
         </value>
       </attr>
     </properties>
  </attributeDefinition>
```

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. Linux 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.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.

Linux Mapping Information

The Linux connector supports the following identify information to be provisioned on the UNIX 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 Linux 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, avoid entering an entitlement (secondary groups) value that is the same value as the Default Group for the system resource. This may cause an entitlement to be inadvertently removed from the user if the user is modified and the Default Group value is changed for that user.

Select Identity Resource Attribute	Linux Attribute	Description
UserName	username	UNIX logon name
Password	password	Logon password
FirstName	F	First Name
LastName	L	Last Name
FullName	comment	Comment section in /etc/passwd
Directory	directory	User's home directory
Shell	shell	UNIX login shell
DefaultGroup	defaultgroup	Default Group membership

3

Understanding the Installed Scripts

The Linux connector uses a Java-based scripting engine called Bean Shell (www.beanshell.org). The Bean Shell scripts control the interactions between the unix machine and the Select Identity server.

There are some special variables and classes available to the Bean Shell script when executing in the Linux connector. Each is described in this chapter.

Special Variables

The following describes the variables used by the installed scripts:

Variable Name	Type	Description
ExpectVerbose	Boolean	If true, server responses are echoed in the output
Login	String	The user name to use when logging on
loginPassword	String	The password to use when logging on
rootPassword	String	The password to use when gaining super user privileges

Variable Name	Type	Description
scriptDir	String	The path to the Select Identity Bean Shell scripts
unixType	String	The type of unix operating system that hostname is running.
Prompt	String	A regular expression to match the unix host prompt.
Args	String	A name value list of extra arguments used by the script.
scriptResponse	String	The output from the last executed script.
Session	Expect Session	The class that provides communication to and from the UNIX host

Special Methods and Functions

These methods and functions enable Select Identity to send and receive data with UNIX systems.

Method Name	Parameters	Description
Session.sendExpect	Pattern	Success case text to expect
	errorPattern	Error case text to expect
	timeout	Timeout in seconds to wait for Pattern
loginExternal	None	Establishes super user privileges
logoutExternal	None	Revokes super user privileges
setPassword	UserId	User to set password for
	OldPassword	Old password for UserId
	NewPassword	New password for UserId

Supplied Scripts

The following scripts are provided during the connector installation:

- `common.bsh`

  This script provides common commands used by several scripts.

- `adduser.bsh`

  This is the script used to add a new user to the Linux system.

- `listusers.bsh`

  This script lists all users configured on the Linux system.

- `changepasswd.bsh`

  This script changes the password of a user on the Linux system.

- `changestatus.bsh`

  This script changes a users status on the Linux system.

- `deleteuser.bsh`

  This script deletes a user from the Linux system.

- `lock.bsh`

  This script defines the lock and unlock functions used by the connector.

- `filelock.sh`

  This script defines a lock and unlock method that the bsh script uses.

- `dotest.bsh`

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

- `finduser.bsh`

  This script is used to search for a user on the Linux system.

- `genericcmd.bsh`

  This script enables the execution of any command required by Select Identity.

- `modifyuser.bsh`

  This script is used to modify a user on the Linux system.

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 using the Select Identity client Connectors pages.

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 Bean Shell scripts as well. Remove the scripts from the directory on the Select Identity server where they were extracted (see Installing Scripts on the Web Application Server on page 9).