HP OpenView Select Identity

3270 Emulation Connector for RACF for OS/390 V2 R10

Installation and Configuration Guide

Connector Version: 3.3
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

This product includes software developed by the Apache Software Foundation (http://www.apache.org/).

Select Identity uses software from the Apache Jakarta Project including:

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

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

Additional third party software used by Select Identity includes:

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

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 3270 Emulation Connector for RACF — hereafter referred to as the RACF connector — enables HP OpenView Select Identity to perform the following tasks in RACF security systems on OS/390 mainframes:

- Add, update, and remove users
- Retrieve user attributes
- Enable and disable users
- Verify a user's existence
- Change user passwords
- Reset user passwords
- Expire user passwords
- Retrieve all entitlements
- Retrieve a list of supported user attributes
- Assign and unassign entitlements to and from users

It is a one-way connector and pushes changes made to user data in the Select Identity database to a target server.

The RACF connector is packaged in the following files:

- `RacfConnector.rar` – contains the resource adapter (connector files)
- `RacfSchema.jar` – contains the mapping file, which controls how Select Identity fields are mapped to RACF table columns, and several example macros

These files are located in the `3270 Emulation for RACF` directory on the Select Identity Connector CD.

System Requirements

The RACF connector is supported in the following environment:

Select Identity Version	Application Server	Database
3.0.2	WebLogic 8.1.2 on Windows 2003	SQL Server 2000
3.3	WebLogic 8.1.4 on Windows 2003	SQL Server 2000
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 RACF security systems on OS/390 or z/OS, version 2 release 10. In addition, the RACF connector uses the IBM SecureWay Host Access Class Library, version 3.0.4-B20000515, to communicate with RACF.

Creating Macros

The RACF connector uses “screen scraping” to establish a session and perform provisioning operations using the 3270 emulator. The commands that are used during the initial logon phase can vary greatly depending on the 3270 emulator used. For this reason, the RACF connector supports the

configuration of a macro that defines logon session request and response sequences. This macro is loaded and executed by the connector at run time to establish the session with server.

You must manually create the macro using a text editor. You can obtain the command sequence for the logon session using any 3270 emulator client against the 3270 server where users will be provisioned by the connector. This is an important step in deploying the RACF connector. If the macro does not execute the actual command sequence, the connector cannot communicate with the 3270 server.

Use an existing 3270 emulator, or download one from an available site, install the emulator, and connect to the system. This chapter does not provide installation instructions for the emulator because the procedure is dependent on the chosen emulator. However, you will need the following information to connect to the system:

- Host name or IP address of the 3270 server
- Port number of the server
- User profile with administrative privileges, which will be used to provision users
- Password of the administrative user

Use this information to establish a session with the 3270 server. The screen should show you options to log on.

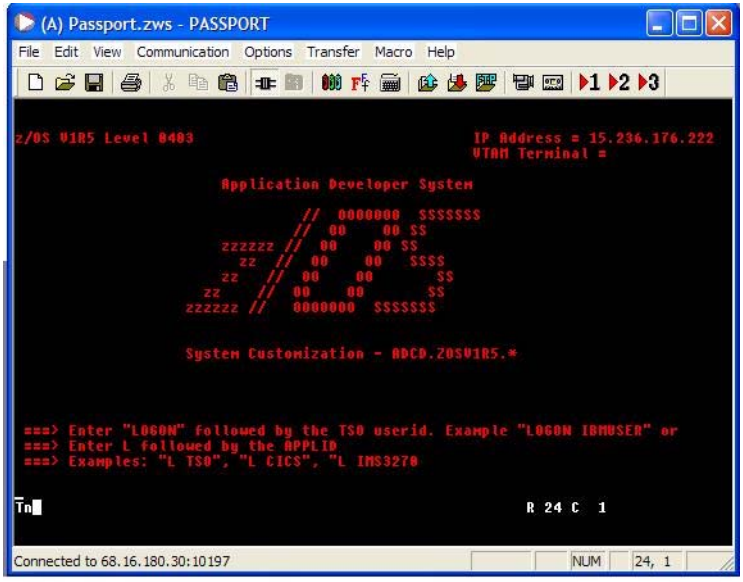
Logon Macro

This macros is required and must contain the sequence of request and response messages to be sent to establish a logon session with the server.

The following are example screens that are used to test the RACF connector.

- Initial Screen

  The administrative user ID is provided on this screen. Here is an example screen that displays when you first connect to the 3270 server:



  This screens displays the following:

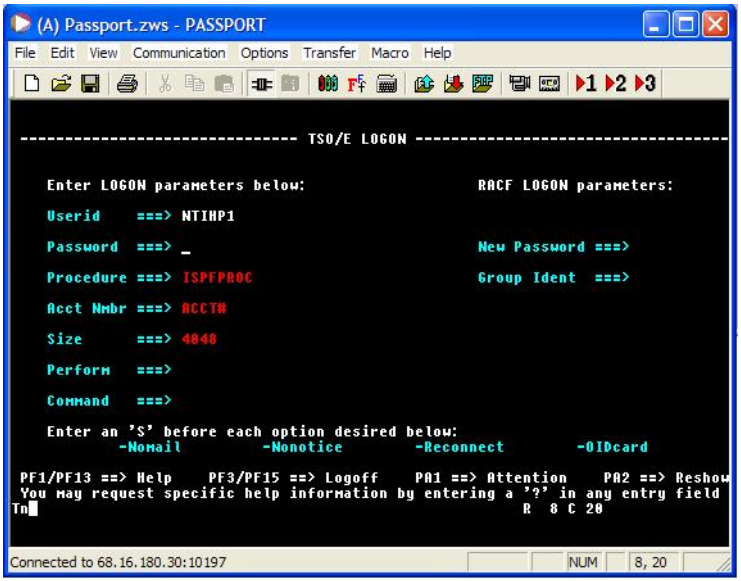
  — Wait for a "LOGON" word after the initial connection

  — Log on to the system by giving the TSO user ID with the LOGON command, as in this example:

    LOGON NTIHP1

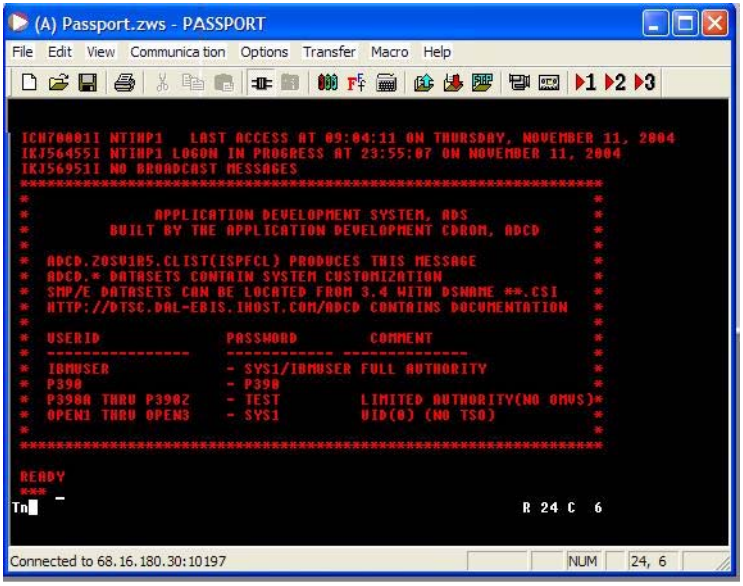
- Password Screen

  The following screen displays after the LOGIN NTIHP1 command is issued in the previous screen. It prompts you for the administrative password:



- Welcome Screen

  After entering the correct password, the system displays a welcome screen, which might look like this:



  This shows the following:

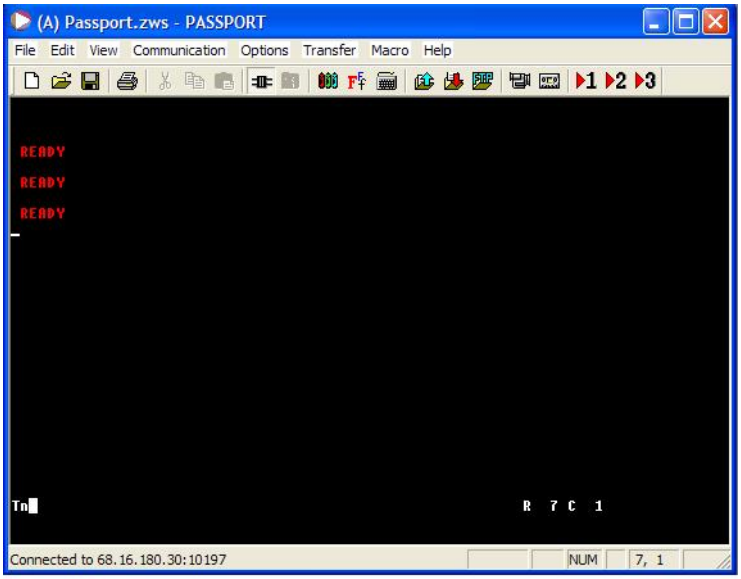
  — The logon was successful

  — The system is ready for additional commands

  — The *** prompt means more input is expected and requires that you to send an ENTER command

- Ready Screen

  The system is ready for your command:



  You must send several ENTER commands to display the System Ready Screen. You can then enter RACF commands.

Based on the screen output, the following is the sequence of steps required to establish a successful session with the 3270 server:

- Wait for LOGON
- Send LOGON <userid> <ENTER>
- Wait for Password ===>
- Send the password
- Send several <ENTER> commands to display the Ready Screen
- Wait for READY by the system

Here is the macro for this session:

```
Wait("LOGON");Send(LOGON ${user}[enter]);
Wait(Password  ===>);Send(${password}[enter]);Delay(1000);
Send([enter]);Send([enter]);Wait(READY
```

Post-creation Macro

Some systems require that a sequence of commands are executed on a newly created user to grant privileges. If your system requires this, the RACF connector can run another macro after creating a new user. The location of this macro can be given in the connection parameters section when deploying the resource.

Here is an example of a post-creation macro, which gives some permissions to the user, such as allowing him to log on to the system:

```
Wait(READY);Send(ALTUSER ${loginUserId} TSO(ACCTNUM(ACCT#)
PROC(ISPFPROC) JOBCLASS(A) MSGCLASS(X) HOLDCLASS(X) SYSOUTCLASS(X)
SIZE(4048) MAXSIZE(0))[enter]);Wait(READY);
Send(PERMIT ACCT# CLASS(ACCTNUM) ID(${loginUserId})[enter]);
Wait(READY);Send(PERMIT ISPFPROC CLASS(TSOPROC)
ID(${loginUserId})[enter]);Wait(READY);Send(PERMIT DBSPROC
CLASS(TSOPROC) ID(${loginUserId})[enter]);Wait(READY);Send(PERMIT
JCL CLASS(TSOAUTH) ID(${loginUserId})[enter]);Wait(READY);
Send(PERMIT OPER CLASS(TSOAUTH) ID(${loginUserId})[enter]);
Wait(READY);Send(PERMIT ACCT CLASS(TSOAUTH)
ID(${loginUserId})[enter]);Wait(READY);Send(PERMIT MOUNT
CLASS(TSOAUTH) ID(${loginUserId})[enter]);Wait(READY);
Send(SETROPTS REFRESH RACLIST(TSOPROC)[enter]);Wait(READY
```

Macro Commands

You can specify the following commands in a macro:

- Wait

  Wait for the occurance of a given string

- Send

  Send the given string to the server

- Delay

  Delyas the macro for a specified number of milliseconds, to synchronize with the server

- Special values

  The following special values can be given in the macro:

  ${user} — Provides the administrative user ID

  ${password} — Sends the administrative user's password

  ${app} — Sends the application name

  Some systems require the application name, such as TSO4, to be sent to the system. This name is sent from the connection parameters:

  ${loginUserId} - Specifies the user ID of the user being created

  [enter] - Send an <ENTER> command

Sample Macros

The `RacfSchema.jar` file is shipped with the following macros:

- `LoginSequence.txt` — Sample shown on page 13
- `PostCreate.txt` — Macro that can be run after users are created
- `SampleLoginSequence_1.txt` — Additional login sample macro
- `SampleLoginSequence_2.txt` — Additional login sample macro

Deploying on the Web Application Server

To install the RACF 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.

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 `RacfConnector.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 16.

   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 `RacfSchema.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 `RacfConnector.rar` file from the list.

   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 `RacfSchema.jar` file (on the Select Identity Connector CD) to the `WebSphere\AppServer\lib\ext` directory.

   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 `RacfConnector.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/Racf`**.

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 21 for details.

After installing the connector, see Configuring the Connector on page 18 for information about registering and configuring the connector in Select Identity.

2

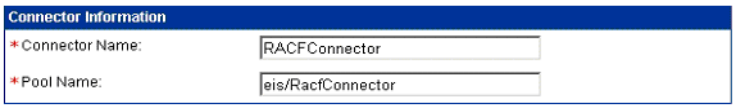
Configuring the Connector

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.

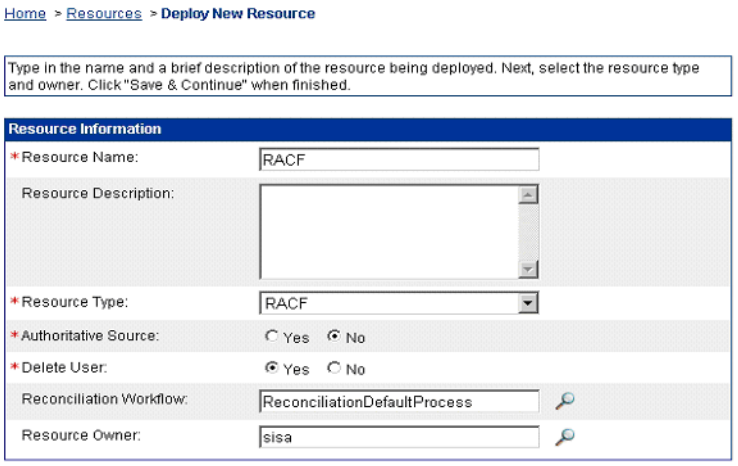
If Test and Submit fails, the 3270 emulator may be active. The 3270 server allows only one logon session at a time per user. If the ID assigned to the connector (in the logon macro) is currently in use, you must first quit the 3270 emulator then retry the resource deployment.

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:

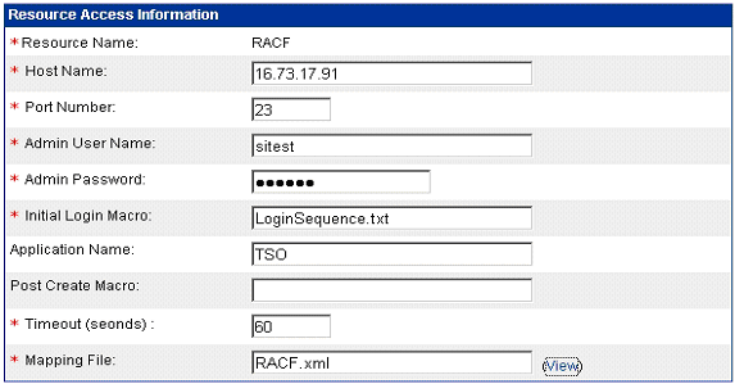


2 Deploy a resource that uses the newly created connector. On the Resources home page, click the **Deploy New Resource** button. Complete the steps in this procedure as described in the "Resources" chapter of the *HP OpenView Select Identity Administrator Guide*. After you deploy the resource for the connector, the Basic Info page of the resource properties will look similar to this:



The Additional Info page of the resource properties will look similar to this:

The Access Info page of the resource properties will look like this:



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.

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 RACF connector is deployed with the `RACF.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 `RacfSchema.jar`. The mapping file is used to map user account additions and modifications from Select Identity to the RACF user table. 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” in RACF.

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 RACF.

  - **<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="RACF 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="User Name"
  required="true" concero:tafield="User Name"
  concero:resfield="UserId" concero:isKey="true"
  concero:init="true"/>
  <attributeDefinitionReference name="Password"
  required="false" concero:tafield="Password"
  concero:resfield="PASSWORD" concero:init="true" />
```

```
    <attributeDefinitionReference name="DefaultGroup"
    required="false" concero:tafield="Default Group"
    concero:resfield="DFLTGRP" concero:init="true" />
    <attributeDefinitionReference name="Owner"
    required="false" concero:tafield="Owner"
    concero:resfield="OWNER" concero:init="true" />
    <attributeDefinitionReference name="Common Name"
    required="true" concero:tafield="'[FirstName] [LastName]'"
    concero:resfield="NAME" concero:init="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 RACF 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 RACF 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. RACF 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 ten characters in length and can contain the following letters, numbers, and characters: a-z, A-Z, 0-9, @, and a +.

  Here is an excerpt from the `RACF.xml` file:

```
<attributeDefinition name="User Name" description="userId"
type="xsd:string" >
    <properties>

      <attr name="minLength">
        <value>1</value>
      </attr>
      <attr name="maxLength">
        <value>10</value>
      </attr>
```

```
        <attr name="pattern">
          <value><![CDATA[[a-zA-Z0-9@]+]]> </value>
        </attr>
...
```

- **<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.

RACF Mapping Information

The RACF connector supports the following identify information to be provisioned on the 3270system. 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 HP-UX. These attributes cannot be changed.

Select Identity Resource Attribute	RACF User Attribute	Description	Mandatory
UserName	USERID	Maximum length is seven characters.	Yes
Password	PASSWORD	Minimum length is eight characters.	No

Select Identity Resource Attribute	RACF User Attribute	Description	Mandatory
Default Group	DFLTGRP	Default group of the user in the system. If not assigned, the system will assign a default group.	No
Owner	OWNER	Owner of the user being created. If not assigned, the administrative user who is creating this user is assigned as the owner.	No
[First Name] [Last Name]	NAME	Full name of the user	No

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.

Uninstalling the Connector from WebLogic

Perform the following to delete a connector:

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

Uninstalling the Connector from 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.