HP OpenView Select Access

For the Windows®, HP-UX®, Linux®, and Solaris® Operating Systems

Software Version: 6.1 and 6.2

Integration Paper for SAP WebAS 6.40

SAP® Certified Integration

Document Release Date: September 2006
Software Release Date: September 2006

hp® invent

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

Trademark Notices

HP OpenView Select Access includes software developed by third parties. The software HP OpenView Select Access uses includes:

- Software developed by the Apache Software Foundation.
- Software developed by Claymore Systems, Inc.
- Cryptographic software written by Eric Young.
- Cryptographic software developed by The Cryptix Foundation Limited.
- cURL, Copyright © 2000 Daniel Stenberg.
- JavaBeans Activation Framework version 1.0.1 © Sun Microsystems, Inc.
- JavaMail, version 1.2 © Sun Microsystems, Inc.
- JavaService software from Alexandria Software Consulting.
- JClass LiveTable, Copyright © 2002 Sitraka Inc.
- The OpenSSL Project for use in the OpenSSL Toolkit.
- Protomatter Syslog, Copyright © 1998-2000 Nate Sammons.
- SAP WebAS 6.40, SAP AG.
- SoapRMI, Copyright © 2001 Extreme! Lab, Indiana University.

For expanded copyright notices, see HP OpenView Select Access `<install_path>/3rd_party_license` directory.

Documentation Updates

This manual's title page contains the following identifying information:

- Software version number, which indicates the software version
- Document release date, which changes each time the document is updated
- Software release date, which indicates the release date of this version of the software

To check for recent updates, or to verify that you are using the most recent edition of a document, go to:

http://ovweb.external.hp.com/lpe/doc_serv/

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Support

You can visit the HP OpenView Support web site at:

www.hp.com/managementsoftware/support

HP OpenView online support provides an efficient way to access interactive technical support tools. As a valued support customer, you can benefit by using the support site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into 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 sign in. Many also require a support contract.

To find more information about access levels, go to:

www.hp.com/managementsoftware/access_level

To register for an HP Passport ID, go to:

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

Contents

1 Understanding Your Select Access Integration

Select Access is an integral part HP's comprehensive Identity Management suite. It delivers a full solution for complex access management across the enterprise. Select Access:

- Automates access control and user life-cycle management
- Extends the enterprise through federation
- Delegates management to business owners and the end users

Along with robust workflow, user self-service, reporting, and delegated administration capabilities, Select Access is the most comprehensive access control system available. Select Access simplifies your ability to secure user access to SAP WebAS Server™ resources.

Assumptions in This Document

This document assumes the following:

- You have SAP WebAS 6.40 with SP 15 or later installed and running on your network.
- You understand the features and functions of Select Access.
- You have installed Select Access 6.1 or 6.2.
- You have a working knowledge of Select Access and LDAP 3.0-compliant directory servers.

Integrating the Select Access and SAP WebAS

In a typical Select Access and EJB application server configuration, all requests for network resources must be sent to the web server. When a request for an EJB application is made via the web server, the Enforcer plugin queries the Policy Validator to determine whether or not the user is allowed to access the resource. If the resource request is authorized according to the access policies set in Select Access, the request proceeds successfully.

Select Access can be installed on the same or different host as the SAP WebAS server.

Understanding the Key Components of this Integration

The key components of the SAP WebAS integration are:

- SALoginModule, the JAAS-based Select Access Enforcer
- Select Access Validator

- LDAP Identity Store

When configured correctly on the WebAS JAAS Login Modules Stack, the SALoginModule Enforcer is invoked when a user accesses an application for the first time. The Enforcer sends an XML query to the Validator. Validator looks up the LDAP Identity Store to find the user, and based on any rules that may be configured, will send an ALLOW or DENY reply. The E Enforcer then lets WebAS know the outcome, and WebAS enforces the decision. If authenticated, the Enforcer places a cookie so the user will not be prompted for a user name and password again during that browser session.

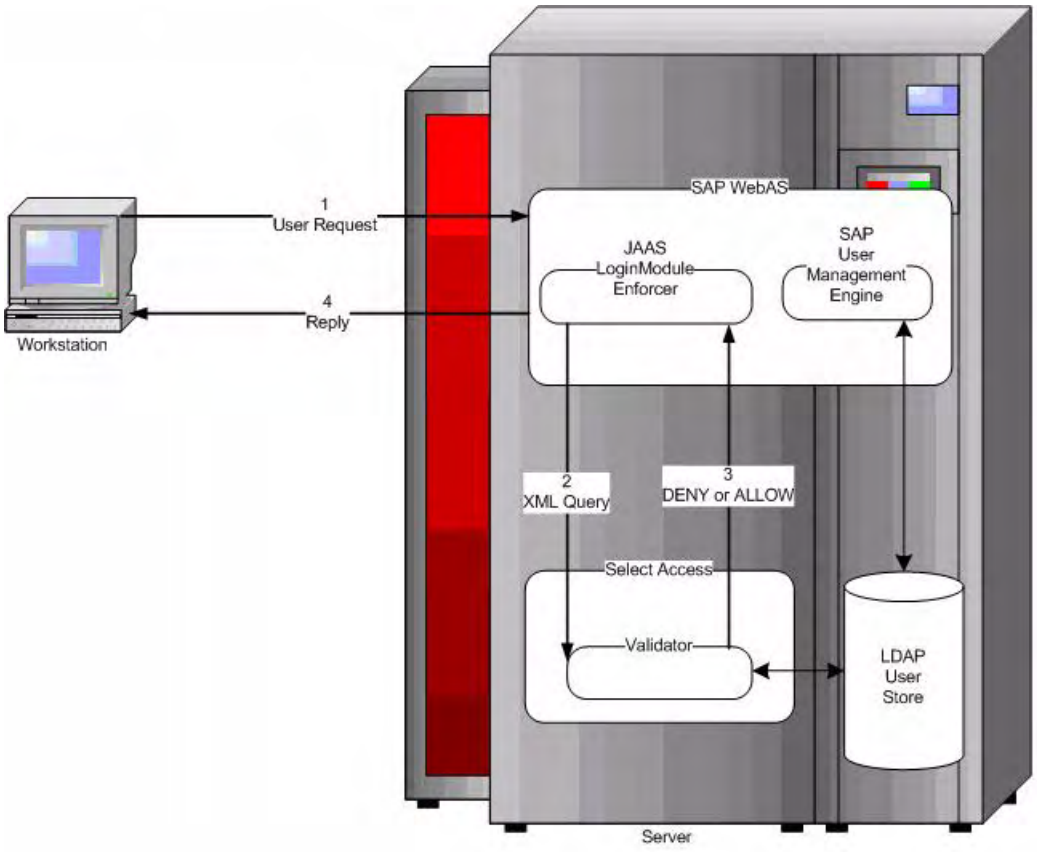

Figure 1 SAP WebAS Components

Limitations

This integration has the following limitations:

- The server is currently unable to return the path of the resource requested by the user. Due to this limitation, resource-level authentication is not possible.
- According to the SAP certification document, there are some limitations of the SAP JAAS implementation due to functionality:
  - Form-based login with POST is not supported
  - Form-based login with URL parameters is not supported

  - The use of HTTP redirects in login modules is only supported for usage types different than the SAP Enterprise Portal. HTTP redirects in login modules may cause problems with user interfaces that use different navigation technologies, e.g. the SAP Enterprise Portal login.

- The SAP WebAS User Management Engine should be configured to use the same LDAP user store as Select Access, unless there is a system in place to synchronize the users in SAP and Select Access.
- Only the following Select Access features are supported:
  - Basic Authentication
  - Personalization

Chapter Summary

This guide includes the chapters listed in Table 1.

Table 1 Chapter Summary

Chapter	Description
Chapter 2, Integration Tasks	This chapter describes the tasks you ned to perform to integrate SAP WebAS with Select Access.
Chapter 3, Accessing Select Access from Your Program	This chapter describes the tasks you need to perform before you can access Select Access.
Chapter 4, Troubleshooting	This chapter contains troubleshooting information and answers to frequently asked questions.

2 Integration Tasks

This chapter describes the tasks you ned to perform to integrate SAP WebAS with Select Access. The tasks are as follows:

- Create an enforcer.xml File
- Create the sa_enforcer.properties File
- Configure SAP WebAS to use LDAP as a Data Source
- Configure Select Access
- Deploy the SDA Archive File
- Configure the Login Module

You must Install Select Access, if it is not already installed, before you begin your integration.

Create an enforcer.xml File

1. Click **Start** → **All Programs** → **HP OpenView** → **Select Access** → **Setup Tool** to run the Select Access Setup Tool.
2. Click **Next** until you reach the **Generic Enforcer Plugin** screen.

You may have to install the Generic Enforcer plugin if it was not installed when Select Access was installed.

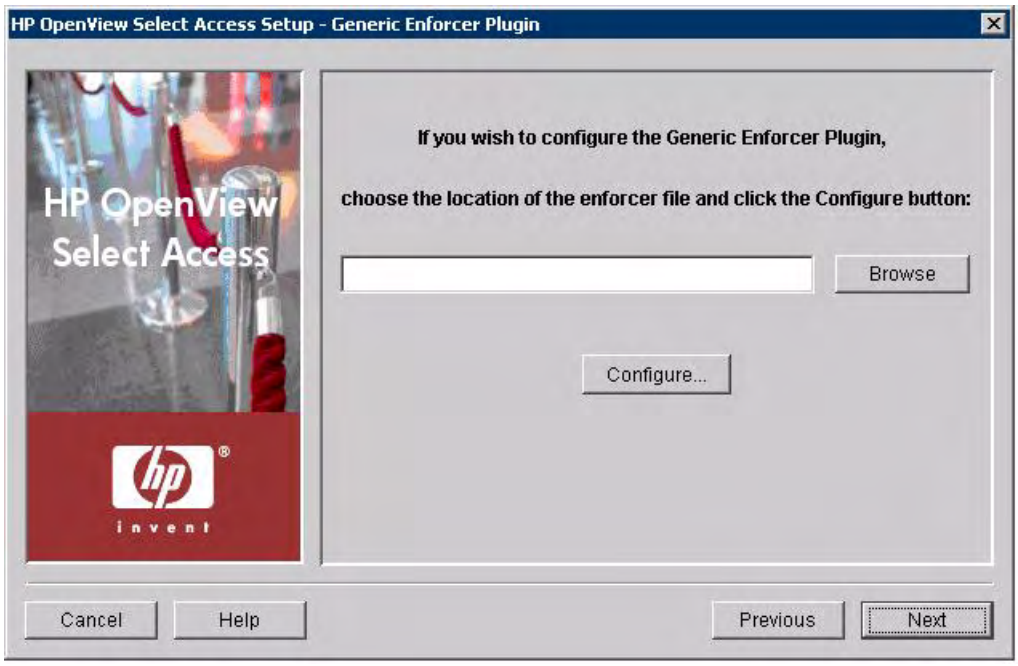

Figure 2 Generic Enforcer Plugin Screen

3 Click **Browse**.

4 Select the directory to store the `enforcer.xml` file. For example, `C:\Program Files\HP OpenView\Select Access\bin\` and click **Choose**.

5 Click **Configure**. The **Contact the Administration Server** screen appears.

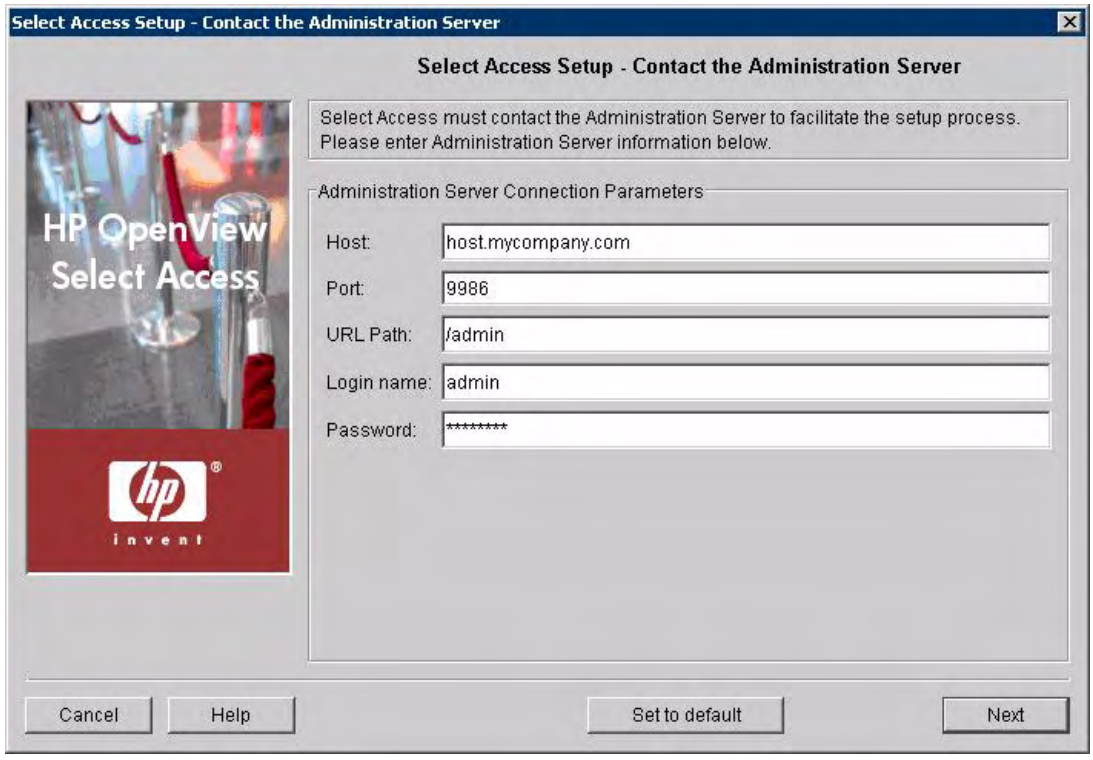

Figure 3 Contact the Administration Server Screen

6 Enter the values required to access your Select Access Admin server and click **Next**. The **Generic Enforcer Plugin Setup - General screen** appears.

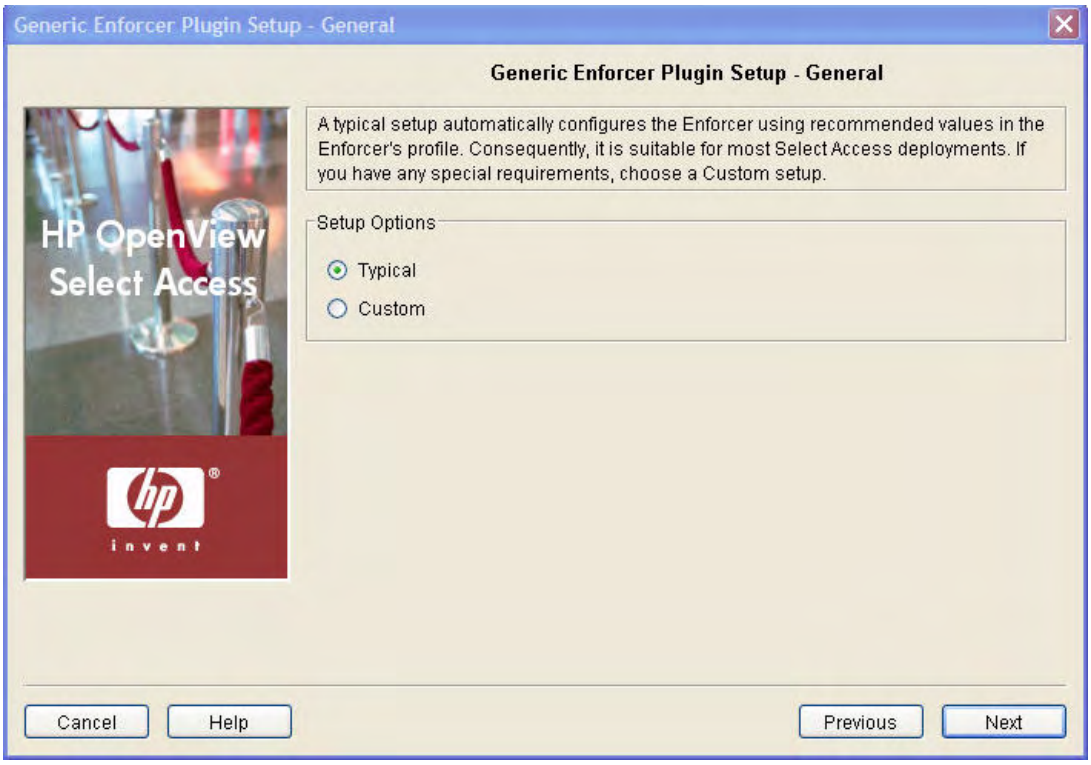

Figure 4 Generic Enforcer Plugin Setup - General Screen

7 Select **Typical** and click **Next**. The **Generic Enforcer Plugin Setup - Finish** screen appears.

8 Click **Finish**.

9 If Select Access and SAP WebAS are being installed on different hosts:

a Copy this `enforcer.xml` file to a directory on the host where WebAS is installed. Note this location which has to be entered in the properties file later on.

b Copy the file `selectaccess.conf` from the Select Access install directory to an identical location in the local machine. For example, under Windows, the directory is `C:\Program Files\HP OpenView\Select Access`.

Create the sa_enforcer.properties File

1 Create a file called `sa_enforecer.properties` under the Select Access installation directory, for example, `C:/Program Files/HP OpenView/Select Access/sa_enforcer.properties`.

If Select Access is installed on a different host, use a directory of your choice. Note this location which has to be entered while configuring the Login Module.

2 Open the `sa_enforcer.properties` file and add the following lines:

```
EnforcerAPIConfigFile=C:/Program Files/HP OpenView/Select Access/bin/enforcer.xml
```

```
Service=http://host.mycompany.com:50000
```

```
Resource=/
SecurityRealm=customRealm
```

- Be sure to use the value appropriate for your installation for `EnforcerAPIConfigFile` (the full path to the `enforcer.xml` file created in Create an enforcer.xml File on page 11) and Service (the URL to access WebAS).
- Use "/", not "\" to separate directories.

Refer to Select Access Enforcer Properties File for SAP WebAS on page 29 for more details.

Configure SAP WebAS to use LDAP as a Data Source

You must configure SAP WebAS to use the same LDAP user store as Select Access.

1. Open a browser and go to `http://help.sap.com`.
2. Click **SAP NetWeaver** on the left panel.
3. Click **Search Documentation**.
4. Enter **LDAP Directory as Data Source** in the **Search** field and click **Search**.
5. Follow the SAP documentation to configure SAP WebAS. Refer to Configuring Sun ONE LDAP Directory as Data Source on page 30 for an example.

   - You can also refer to SAP note 673824.
   - If you have problems changing the password of an LDAP user after this procedure, please refer to SAP note 868194 entitled *Change password for LDAP users is not working*.

Configure Select Access

You must configure Select Access to work with SAP WebAS

1. Click **Start** → **All Programs** → **HP OpenView** → **Select Access** → **Policy Builder**.
2. Add your server to the **Resource Access** list.
3. In the Policy Builder **Resource Server**, right click and add a folder called **http**.
4. Right click and add **New Resource Server** to the **http** folder.

5 Enter the name of the Resource Server, the protocol, hostname and port number. For example:

protocol: `http`

hostname: `saintw01.can.hp.com`

port #: `50000`

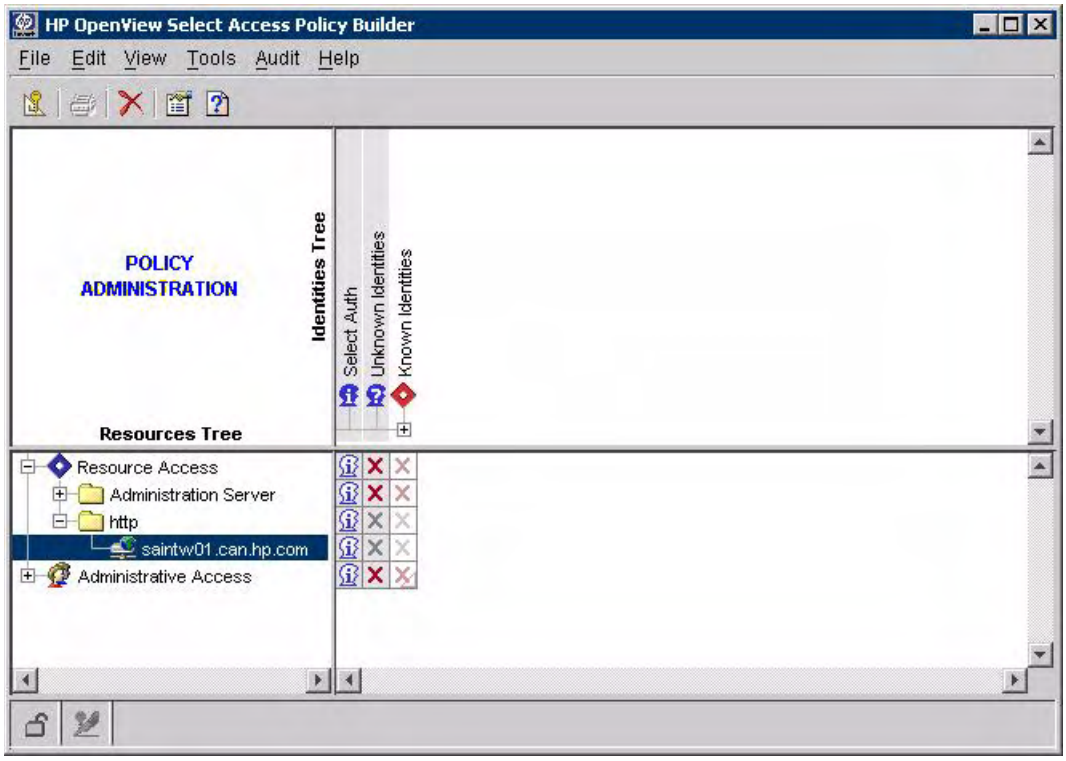

Figure 5 Select Access Policy Builder

6 Right click the **Select Auth** icon corresponding to your server and click **Enable Select Auth**.

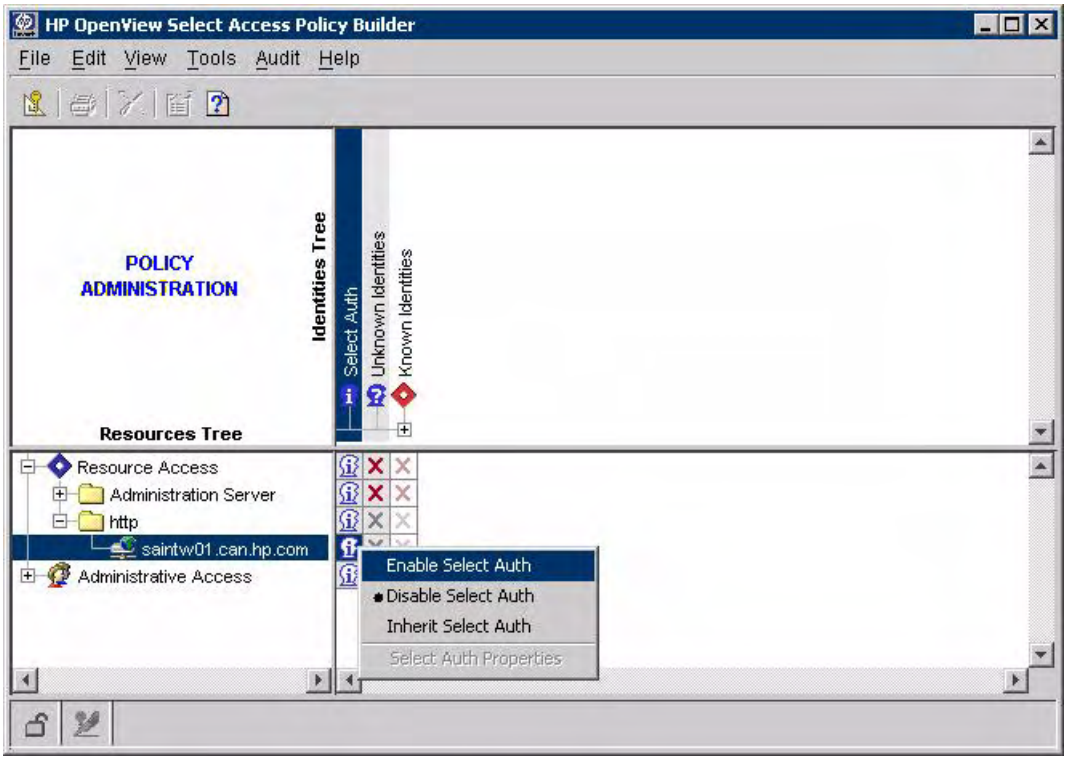

Figure 6 Enable Select Auth

The **Authentication Properties** dialog box appears.

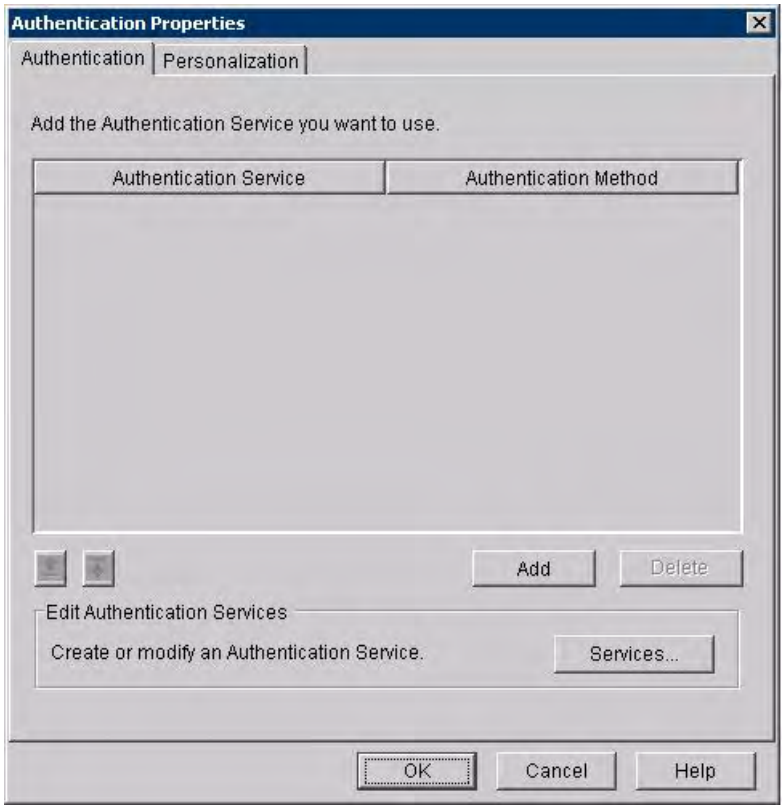

Figure 7 Authentication Properties Dialog Box

7 Click **Services**. The **Authentication Services** dialog box appears.

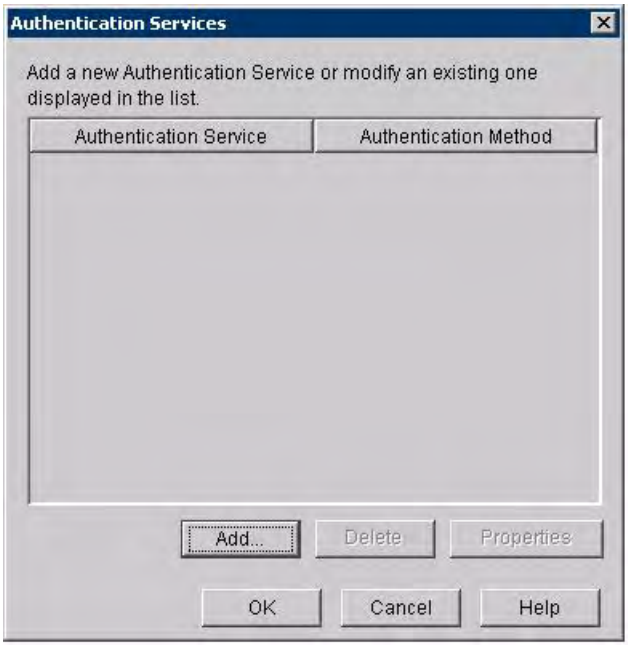

Figure 8 Authentication Services Dialog Box

8 Click **Add**. The **New Authentication Service** dialog box appears.

9 Select **Password** and enter Password in the **Service Name** field.

Figure 9 New Authentication Service Dialog Box

10 Click **OK** three times to return to the **Authentication Properties** dialog box.

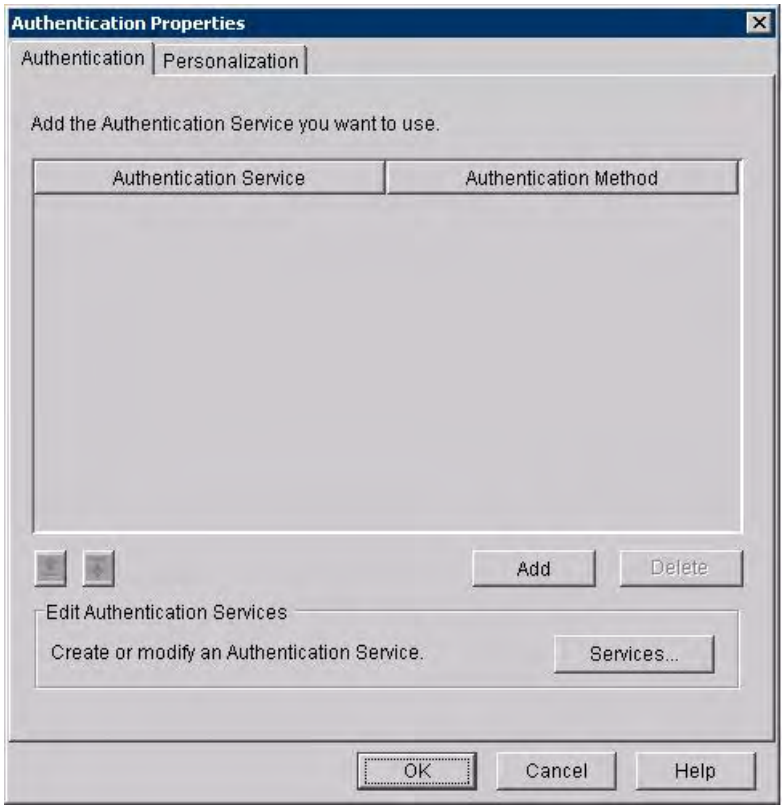

Figure 10 Authentication Properties Dialog Box

11 Click **Add**. The **Available Authentication Services** dialog box appears.

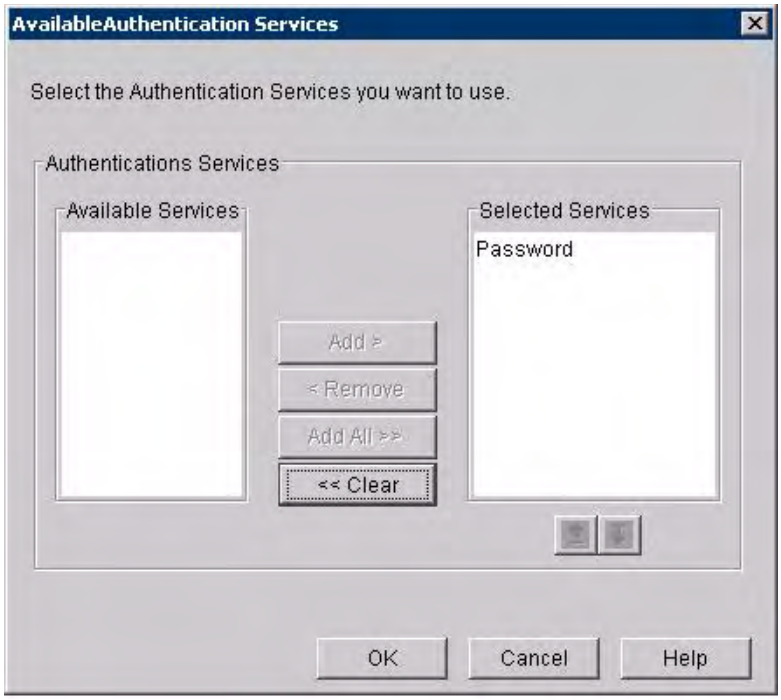

Figure 11 Available Authentication Services Dialog Box

12 Select **Password** from the **Available Services** list and click **Add** to move it to the **Selected Services** list.

13 Click **OK** to return to the **Authentication Properties** dialog box.

14 Clicking the **Personalization** tab. The **Authentication Properties Personalization** tab is displayed.

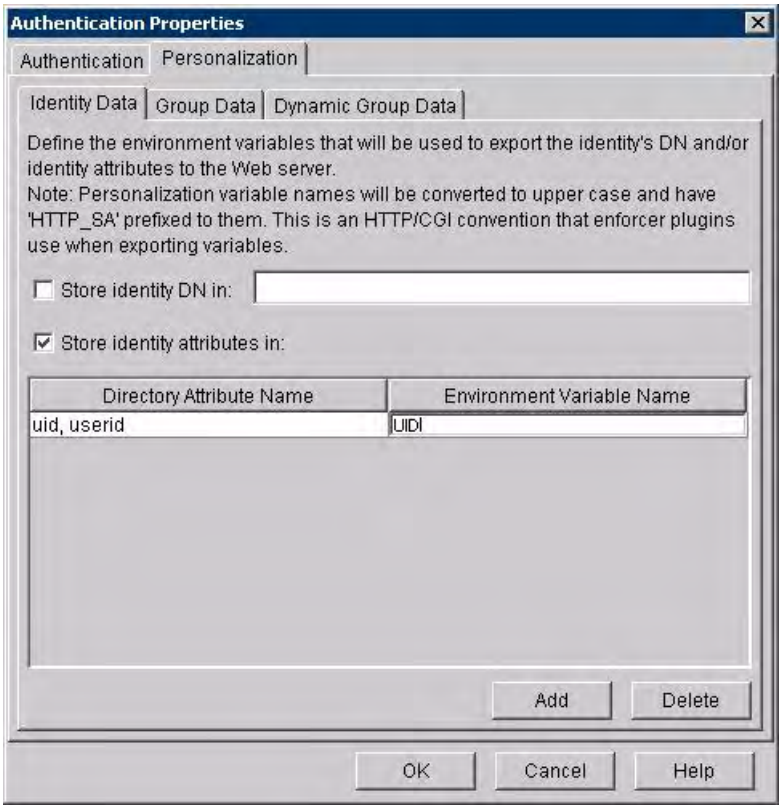

Figure 12 Authentication Properties Personalization Tab

15 Select the **Store identity attributes in** check box.

16 Click **Add** and select `uid, userid` from the **Directory Attribute Name** drop-down list.

17 Enter `UID` in the **Environment Variable Name** text box.

18 Click **OK**.

Configure Known Identities

1 Right-click **Known Identities** in the Policy Builder and select **New** → **Identity Location**.

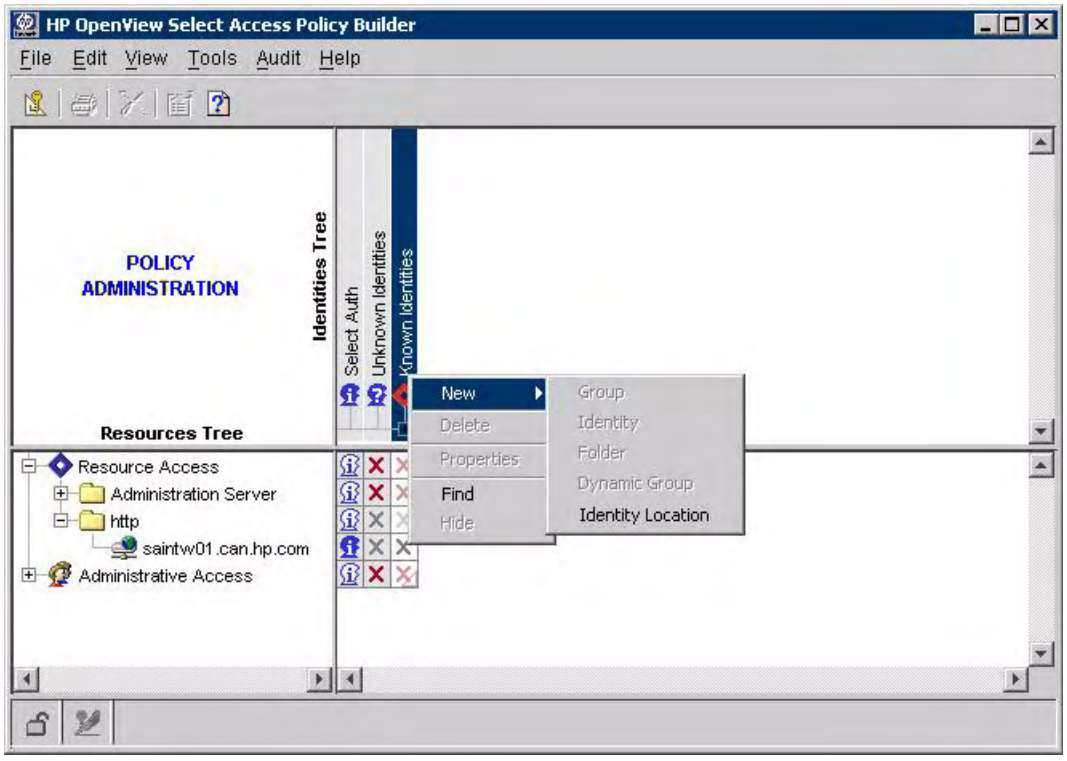

Figure 13 Policy Builder New Identity Location

The **New Identity Location** dialog box appears.

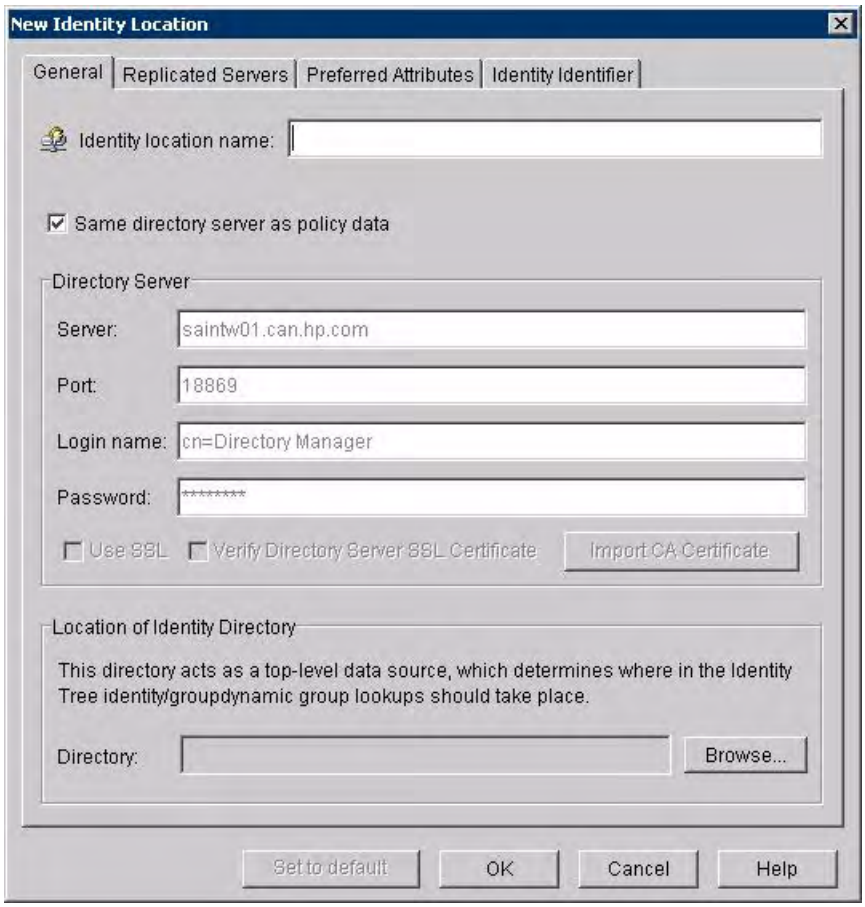

Figure 14 New Identity Location Dialog Box

2 Enter a name in the **Identity location name** field. Confirm that the correct values are entered for the LDAP directory server.

3 Click **Browse** to select the LDAP location. The **LDAP Location** dialog box appears.

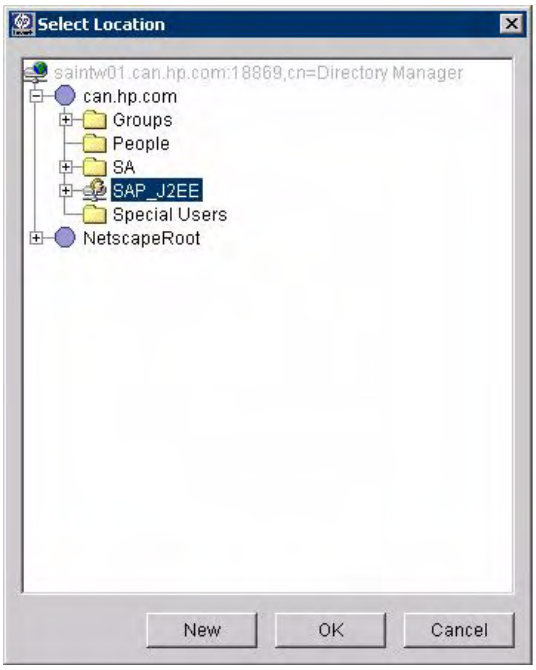

Figure 15 LDAP Location Dialog Box

4 Select a location and click **OK**. The location is displayed in the **Directory** field of the **New Identity Location** dialog box.

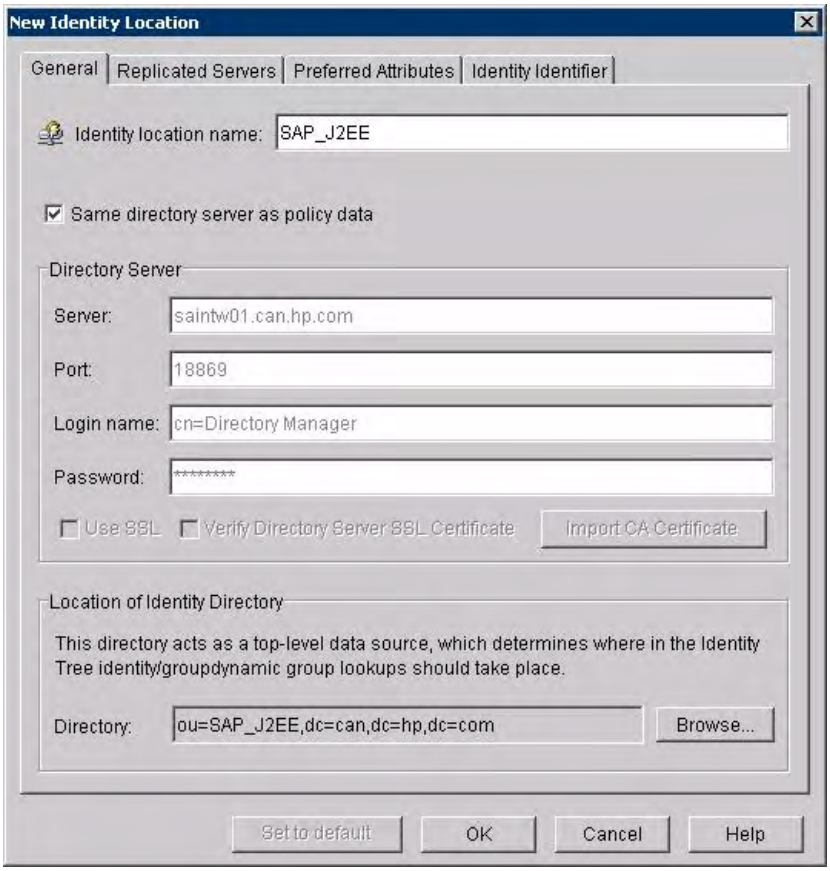

Figure 16 New Identity Location Dialog Box

5 Click **OK**.

6 Add new identities using the **SAP User Management** screen and refresh the Select Access screen for users to appear in the list.

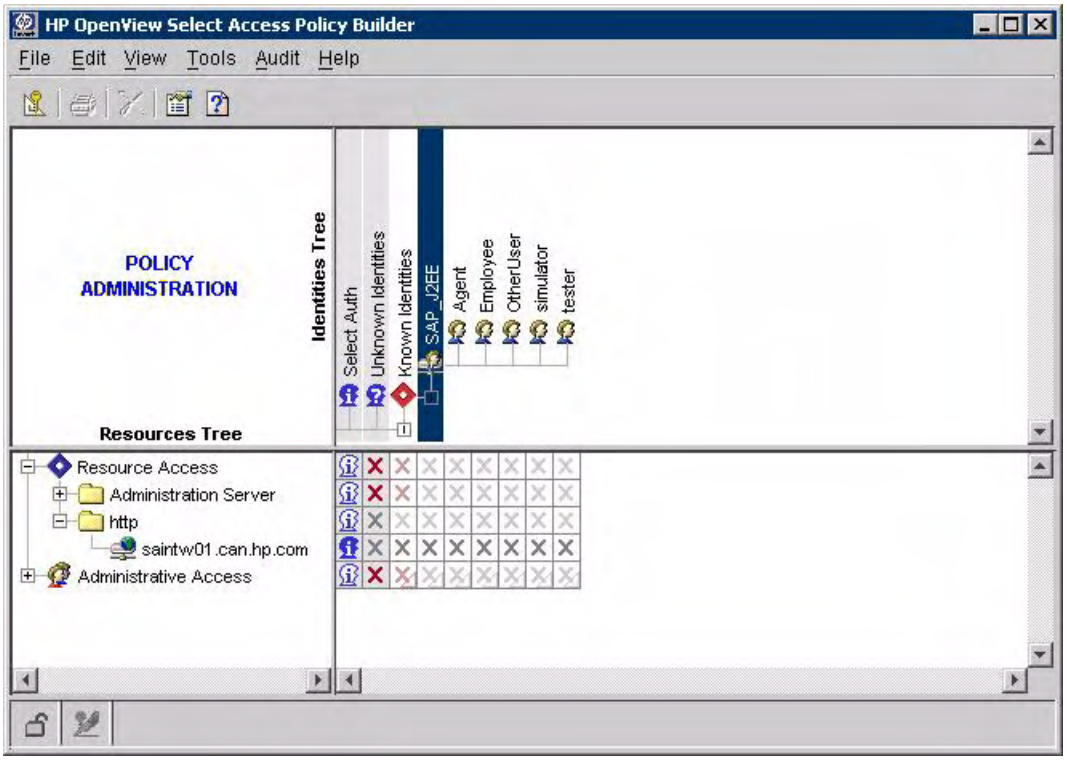

Figure 17 Policy Builder SAP Users

Due to limitations in SAP, users created before LDAP was set up as the user store may not appear in the list. In such cases, delete and then add the user.

Deploy the SDA Archive File

You now need to deploy the archive file `SALoginModuleSAPLibrary.sda` using the SAP Software Deployment Manager GUI. This file should be available on the install CD.

1 Run the batch file *<SAP Install Directory>*`\usr\sap\`*<SID>*`\JC`*<SAP instance>*`\SDM\program\RemoteGui.bat`.

For example: `E:\usr\sap\TWD\JC00\SDM\program\RemoteGui.bat`.

2 Log in to the server. The **SAP Software Deployment Manager** appears.

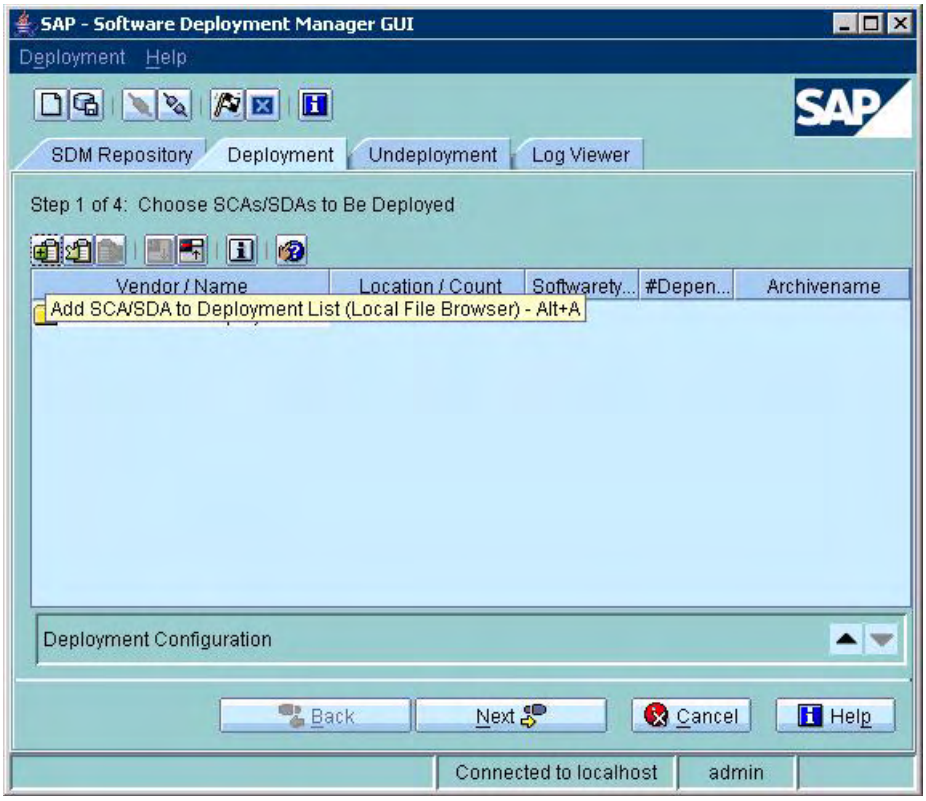

Figure 18 SAP Software Deployment Manager

3 Click the **Deployment** tab.

4 Click the **Add SCA/SDA** icon or press `Alt + A`. The ****What is its name??****dialog box appears.

5 Select the SDA file provided and click **Choose**. The archive gets loaded as shown in Figure 19.

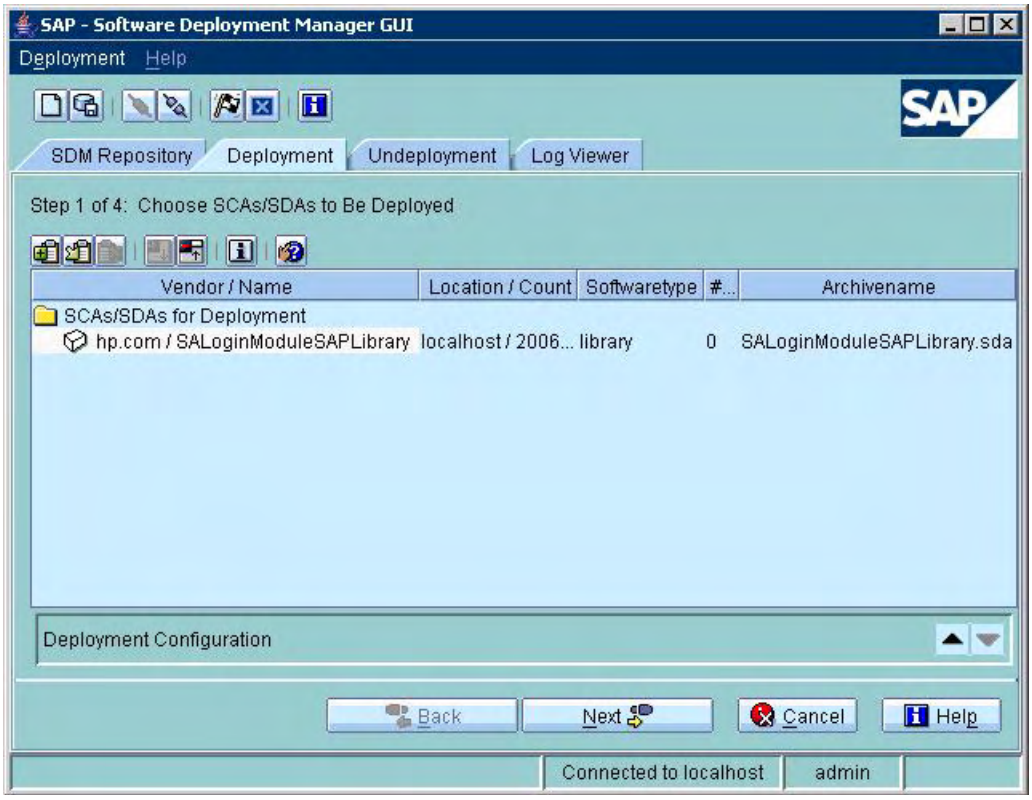

Figure 19 SDA Deployment

6 Click **Next** twice and then click **Start Deployment**.

7 When the deployment is finished, click **Confirm**.

It may be necessary to restart the server after deployment.

Configure the Login Module

The Login Module is configured using SAP Visual Administrator.

1 Run the batch file *<SAP Install Directory>*`\usr\sap\`*<SID>*`\JC`*<SAP instance>*`\j2ee\admin\go.bat`.

For example: `E:\usr\sap\TWD\JC00\j2ee\admin\go.bat`.

2 Log in as `Administrator`.

3 Click **Server...** → **Services** → **Security Provider** in the left panel, as shown in Figure 20.

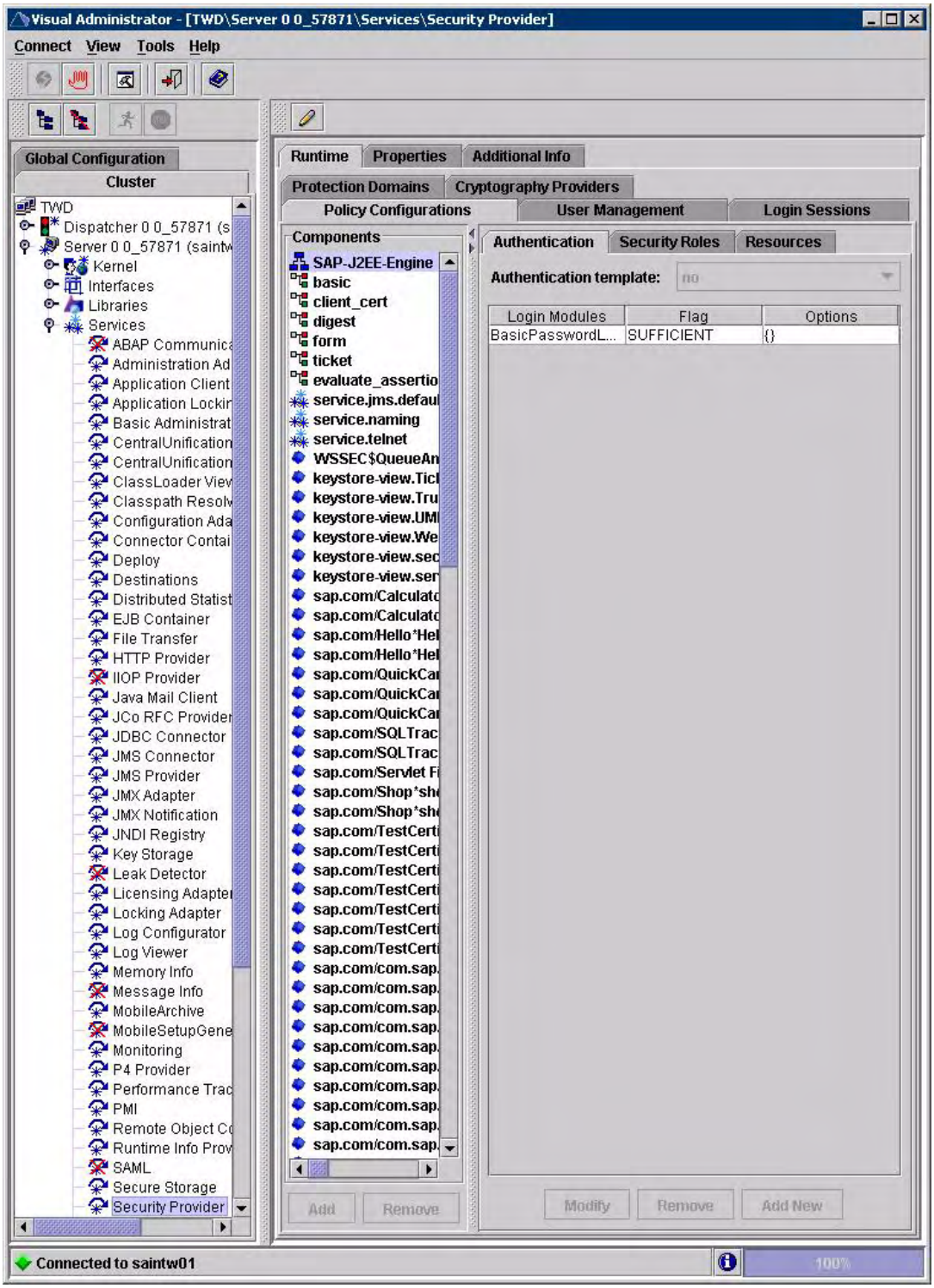

Figure 20 Security Provider

4 Click the **Properties** tab.

5 Click **LoginModuleClassLoaders**.

6 Enter `library:hp.com~SALoginModuleSAPLibrary` in the **Value** text box at the bottom and click **Update**.

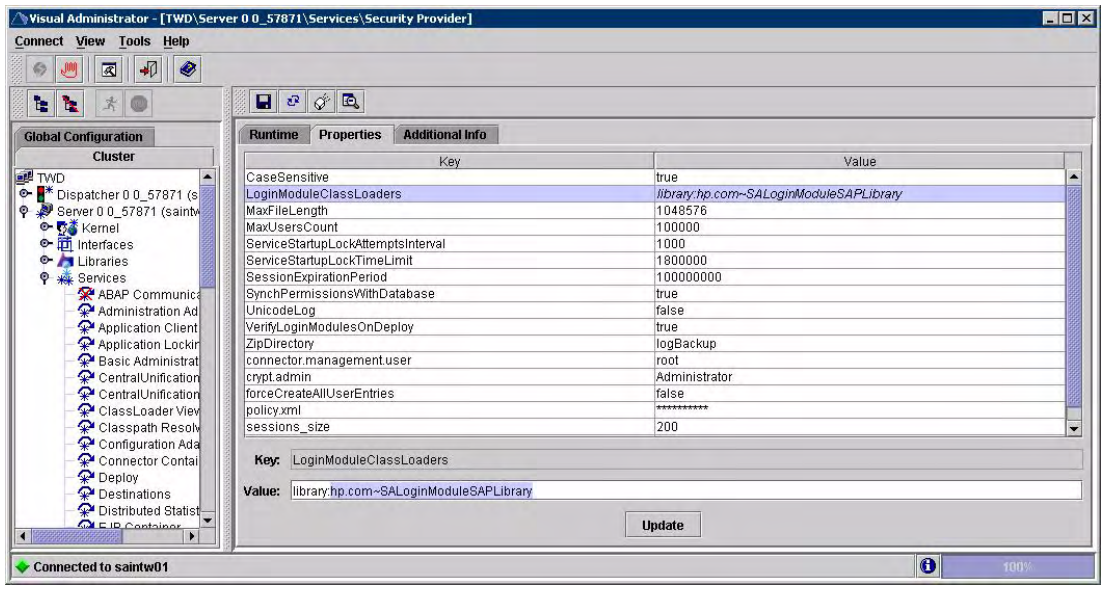

Figure 21 Security Provider Properties

7 Click the **Runtime** tab and then click **User Management**.

8 Click the **Pencil Icon**  at the top of the Security Provider to switch to **Edit** mode.

9 Click **Manage Security Stores** then click **Add Login Module**. The **Edit Login Module** dialog box appears.

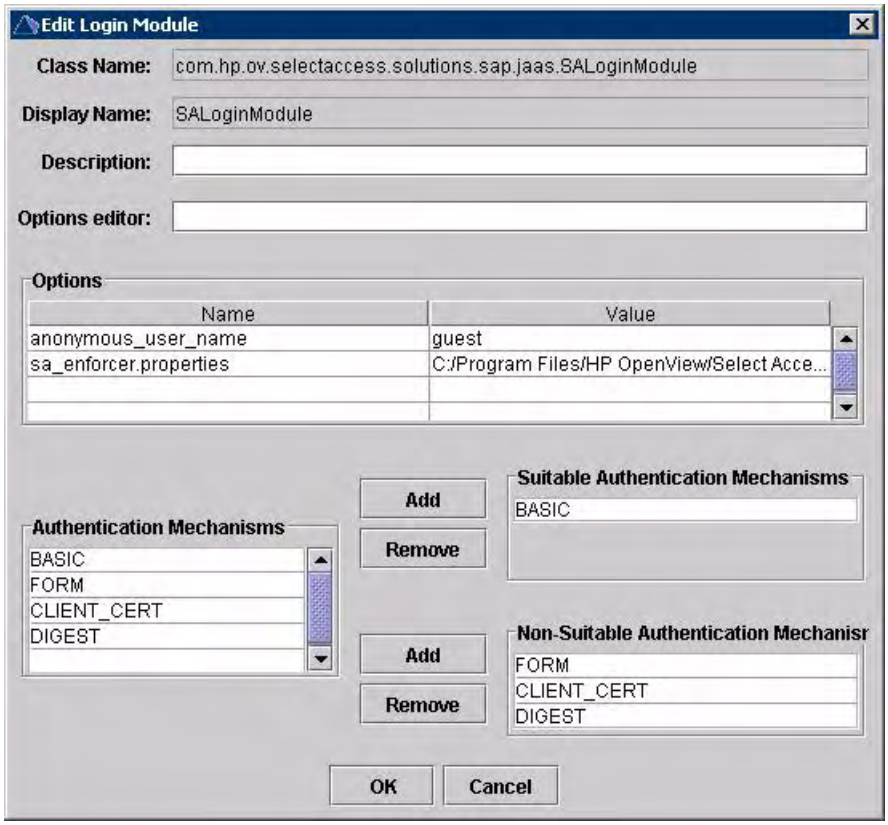

Figure 22 Edit Login Module Dialog Box

10 Enter the following information.

Class Name: `com.hp.ov.selectaccess.solutions.sap.jaas.SALoginModule`

Display Name: SALoginModule

11 Enter the following in the **Name** and **Value** columns.

Name	Value
`anonymous_user_name`	guest
`sa_enforcer.properties`	*<Select Access install directory>*/`sa_enforcer.properties` e.g. `C:/Program Files/HP OpenView/Select Access/sa_enforcer.properties`

- The value for `anonymous_user_name` should be the user ID to be used for anonymous access.
- Use "/" , not "\" to separate directories.

12 From the **Authentication Mechanisms** list, select **BASIC** and click **Add** beside the **Suitable Authentication Mechanism** field.

13 From the **Authentication Mechanisms** list, select **FORM**, **CLIENT_CERT**, and **DIGEST** and click **Add** beside the Non-**Suitable Authentication Mechanism** field.

14 Click **OK**

15 Click the **Policy Configurations** tab and select the application you would like to secure.

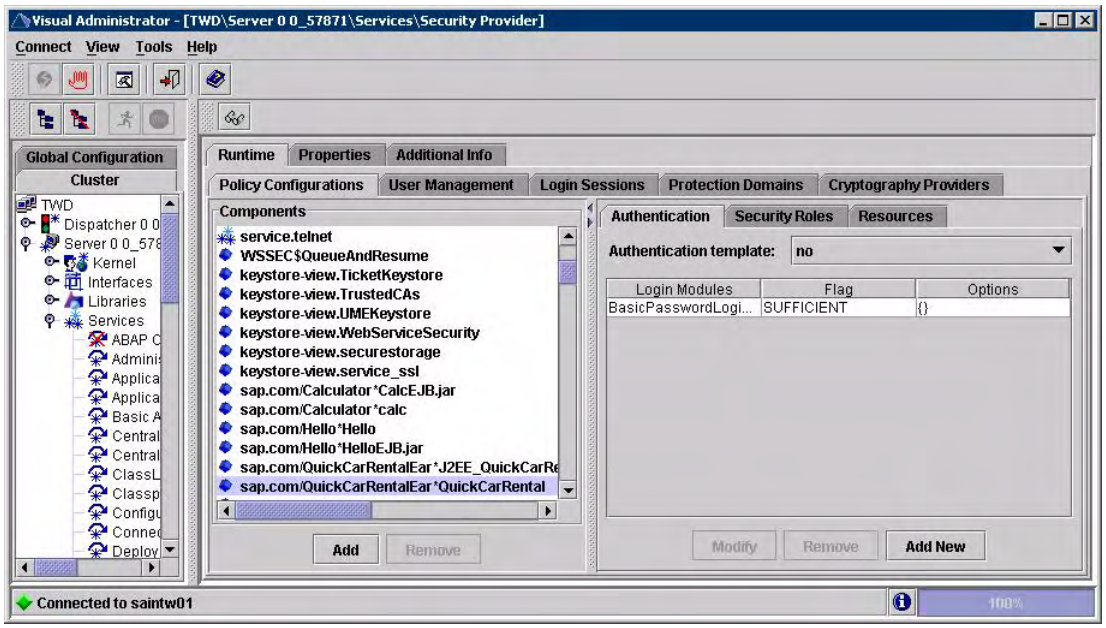

Figure 23 Security Provider Policy Configurations

16 Click **Add New**. The **Available Login Modules** dialog box appears.

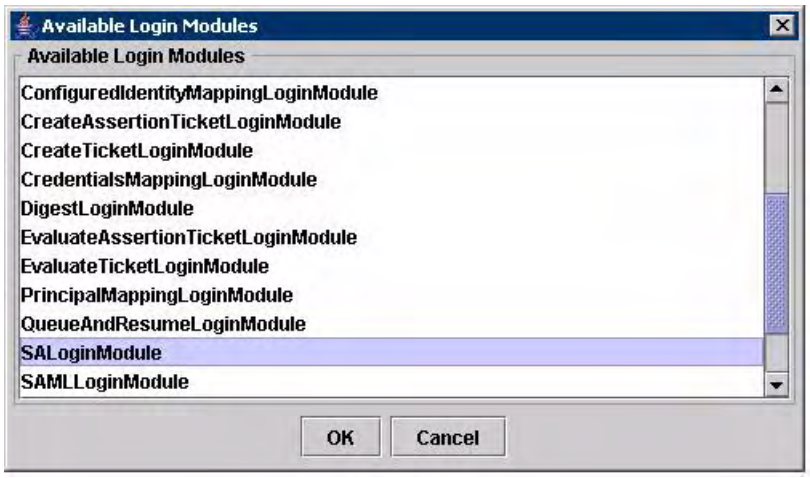

Figure 24 Available Login Modules

17 Select **SALoginModule** and click **OK**.

Select Access Enforcer Properties File for SAP WebAS

A typical properties file contains entries similar to the following:

```
EnforcerAPIConfigFile=C:/Program Files/HP OpenView/Select Access/bin/
enforcer.xml

Service=http://host.mycompany.com:50000

Resource=/

SecurityRealm=customRealm
```

These parameters are described in Table 2.

Table 2 Properties File Parameters

Parameter	Value	Description
`EnforcerAPIConfigFile`	The full path to the `enforcer.xml` file	Created by the Setup Tool when using the Generic Enforcer setup. It contains information on how to connect to the Policy Validator from the Enforcer plugin.
`Service`	The URL to the server e.g.: `http://host.mycompany.com:50000`	This specifies the default protocol, hostname and port used, if the Enforcer is unable to retrieve them from the server.
`Resource`	`/`	Always specify "/" since resource level protection is not supported
`SecurityRealm`	`customRealm`	This value is not used by this Enforce but is required by Select Access.

Configuring Sun ONE LDAP Directory as Data Source

This is a sample setup for WebAS 6.40 SP15. You may use your own choice of LDAP server.

1 Start the SAP J2EE Engine Config Tool using the batch file:

`<SAP Install Directory>\usr\sap\<SID>\JC<SAP instance>\j2ee\configtool\configtool.bat`.

For example, `E:\usr\sap\TWD\JC00\j2ee\configtool\configtool.bat`.

The **SAP J2EE Engine Config Tool** appears.

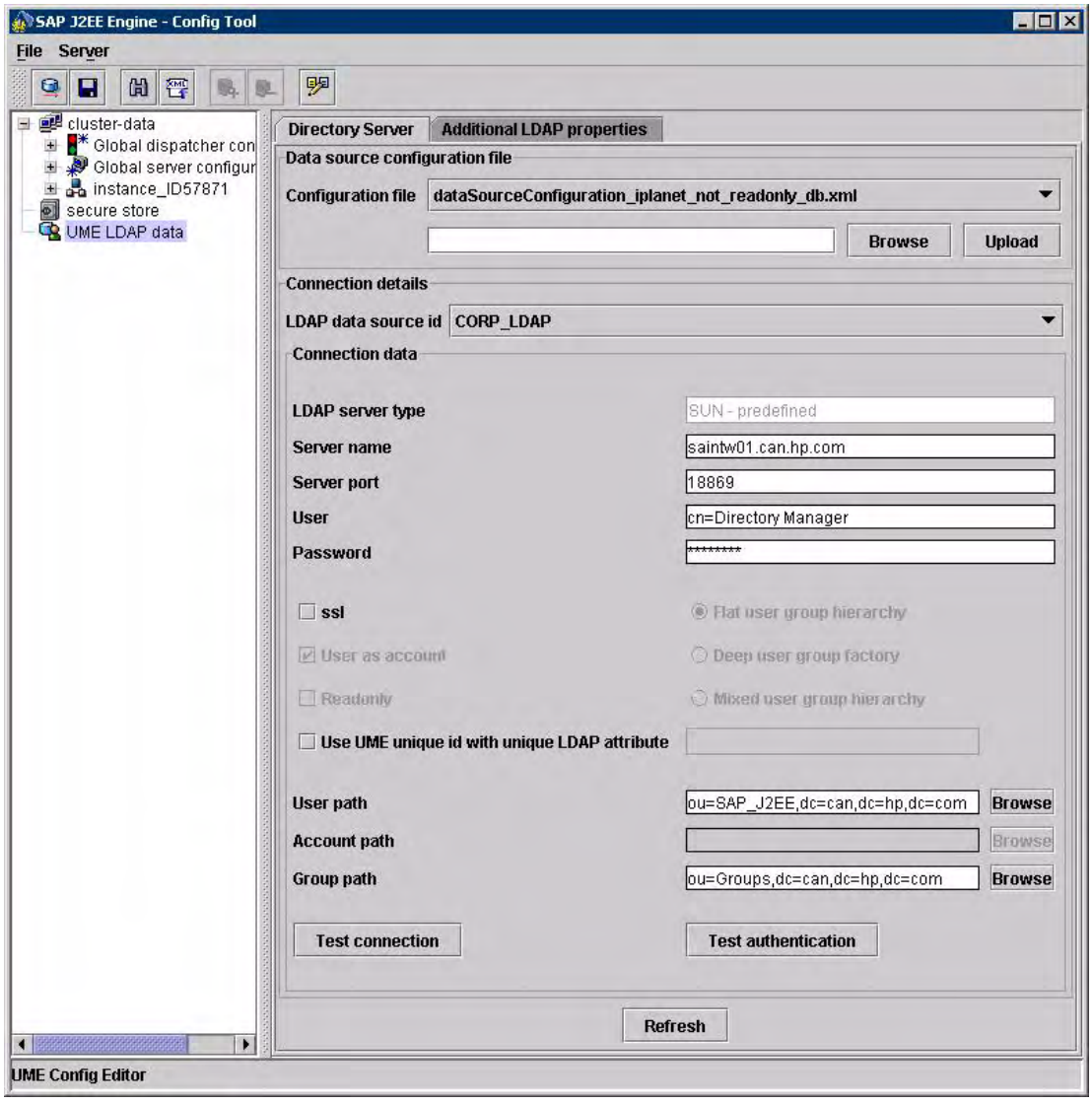

Figure 25 SAP J2EE Engine Config Tool

2 Select **UME LDAP data** in the left panel.

3 Select the `dataSourceConfiguration_iplanet_not_readonly_db.xml` file from the **Configuration file** from the drop-down box.

4 Enter the connection information for your installation.

In the **Server Name** field, enter the same server as you used for the Select Access installation.

3 Accessing Select Access from Your Program

In order to access the Select Access API, you must do the following:

- Extract the necessary JAR files.
- Add the JAR files to the build path of your IDE.

To add JAR Files

1. Extract the JAR files from `SALoginModuleSAPLibrary.SDA` that was distributed to you for the installation.
2. In your `application-j2ee-engine.xml` file, add a reference to the external library `hp.com~SALoginModuleSAPLibrary`:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE application-j2ee-engine SYSTEM
"application-j2ee-engine.dtd">
<application-j2ee-engine>
  <reference
    reference-type="hard">
    <reference-target
      provider-name="hp.com"
      target-type="library">SALoginModuleSAPLibrary
      </reference-target>
  </reference>
  <fail-over-enable
    mode="disable"/>
</application-j2ee-engine>
```

Accessing Personalization Data from an EJB

The code sample below shows how to access personalization data from an EJB. This is not a complete program. Only relevant portions of the code are shown here. The QuickCarRental example code provided by SAP is used here.

```
import com.hp.ov.selectaccess.solutions.sap.jaas.Principal;
public class QuickOrderProcessorBean implements SessionBean {
   private SessionContext myContext;
   public void setSessionContext(SessionContext context) {
       myContext = context;
   }
   public QuickBookingModel saveBooking(
```

```
        java.security.Principal javaPrincipal = myContext.
        getCallerPrincipal();
        if (javaPrincipal instanceof com.hp.ov.selectaccess.solutions.
        sap.jaas.Principal){
            String p13NvalueUID = ((Principal)javaPrincipal).getP13n().
            getAttribute("UID");
            System.err.println("P13N value of UID is :" + p13NvalueUID);
            String p13NvalueDN = ((Principal)javaPrincipal).getP13n().
            getAttribute("DN");
            System.err.println("P13N value of DN is :" + p13NvalueDN);
        }else{
            System.err.println("Principal is not a select access
            principal");
        }
    }
}
```

Accessing Personalization Data from a JSP Page

The code sample below shows how to access personalization data from a JSP page.This is not a complete program. Only relevant portions of the code are shown here. The QuickCarRental example code provided by SAP is used here.

```
<%@ page import="com.hp.ov.selectaccess.solutions.sap.jaas.Principal"
language="java" %>
<form name="wizard" method="POST" action="/QuickCarRental">
<%
  out.println("<BR>Principal is: " + request.getUserPrincipal() +
"<BR>");
  java.security.Principal javaPrincipal = request.getUserPrincipal();
  if (javaPrincipal instanceof com.hp.ov.selectaccess.solutions.sap.
  jaas.Principal){
      String p13NvalueUID = ((Principal)javaPrincipal).getP13n().
      getAttribute("UID");
      out.println("<BR>P13N value of UID is :" + p13NvalueUID + "<BR>");
      String p13NvalueDN = ((Principal)javaPrincipal).getP13n().
      getAttribute("DN");
      out.println("<BR>P13N value of DN is :" + p13NvalueDN + "<BR>");
  }else{
      out.println("<BR>Principal is not a select access principal<BR>");
  }
%>
</form>
```

4 Troubleshooting

Checking the Logs

There are two logs you can use to troubleshoot your integration. If the login module is not able to communicate with the Validator, you can check the SAP logs. If the communication is working, you can check the Select Access logs. Refer to the *HP OpenView Select Access 6.2* documentation for more information about Select Access logs.

To check the SAP Logs

1. Start the **SAP J2EE Visual Administrator** using the batch file *<SAP Install Directory>*\usr\sap*<SID>*\JC*<SAP instance>*\j2ee\admin\go.bat.

   For example, `E:\usr\sap\TWD\JC00\j2ee\admin\go.bat`.

2. Log in as **Administrator** and click **Server...** → **Services** → **Log Viewer**.
3. Select **j2ee\cluster\server0** → **log** → **defaultTrace.trc** as shown in Figure 26.

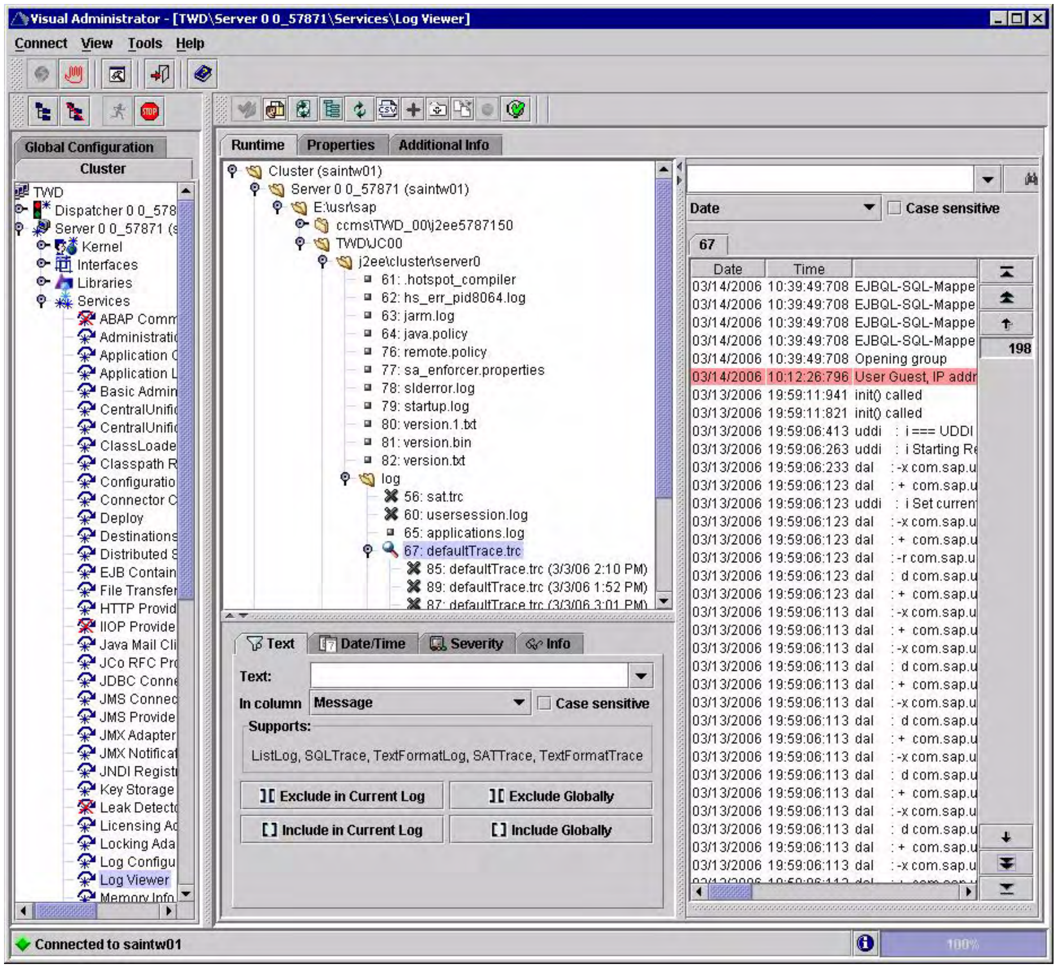

Figure 26 Log Viewer

Frequently Asked Questions

Q--->Why does my user name or user ID come up as something long such as cn=JoeDoe,dc=can,dc=hp,dc=com?

A--->You need to configure a personalization parameter named UID with the LDAP attribute uid. Refer to Step 14 on page 19 for information about setting the personalization parameter.

Q--->Why is the user not authenticated even when Select Access returns an ALLOW?

A--->SAP expects the principal to be given the same name as the user ID. This is not possible if personalization is not set up as described above. You need to configure a personalization parameter named UID with the LDAP attribute uid as explained in Configure Select Access on page 14.

Q--->Why is the user always authenticated as Guest without being asked for user name and password?

A--->Unknown Identities are being allowed access in Select Access. Disable access to Unknown Identities in Select Access if you would like the user to be prompted for user name and password.

Index