

HP OpenView Select Access

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

Software Version: 6.2, 6.1 and 6.0 (with Patches)

Integration Paper for Select Identity

Document Release Date: September 2006

Software Release Date: September 2006



Legal Notices

Warranty

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

The information contained herein is subject to change without notice.

Restricted Rights Legend

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

Copyright Notices

© Copyright 2004-2006 Hewlett-Packard Development Company, L.P.

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.
- SoapRMI, Copyright 2001 Extreme! Lab, Indiana University.

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

WebLogic Server is a US trademark of BEA Systems, Inc.

WebSphere is a US trademark of IBM, Inc.

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

HP-UX Release 10.20 and later and HP-UX Release 11.00 and later (in both 32 and 64-bit configurations) on all HP 9000 computers are Open Group UNIX 95 branded products.

Intel®, Pentium®, and Itanium® are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux® is a U.S. registered trademark of Linus Torvalds.

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

Unix® is a registered trademark of The Open Group.

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 HP OpenView Identity Management	9
	Assumptions in this Document	9
	Integrating the Select Access and Select Identity Frameworks	9
	Understanding the Two Architectures	10
	Understanding Critical Integration Points	10
	Example of an Integrated Network Architecture	11
	The Select Identity User Authentication Process	11
	Integration Task Overview	12
2	Deploying the HP OpenView Identity Management Solution	13
	Deploying the Select Access Enforcer Plugin for BEA WebLogic	14
	Deploying Select Access on a BEA WebLogic Host	14
	To install an Enforcer plugin	15
	To configure the plugin and create the XML bootstrap file	15
	Defining Select Access Classpaths	16
	To append Enforcer API JAR files to the classpath	16
	Storing WebLogic Connection Information	18
	To create the bea_enforcer.properties file	18
	Making Select Access the WebLogic Security Provider	20
	To set up Select Access as a WebLogic Security Provider	20
	Setting Default Security Constraints for WebLogic Resources	21
	To configure the WebLogic server to protect all resources	21
	Enabling Form-based Authentication on WebLogic	22
	To set up SSO in Select Identity	23
	Using Graphics on Login Pages	25
	Deploying IBM WebSphere with the IBM HTTP Proxy & the Select Access Enforcer Plugin	26
	To disable fast response caching	26
	To deploy the IBM HTTP Enforcer plugin	26
	Reconfiguring Your Select Identity System	27
	To reconfigure Select Identity for WebLogic and WebSphere deployments	28
3	Protecting Select Identity Assets	29
	Registering Select Identity Resources	29
	Defining the Select Identity Services	29
	To add Select Identity's application server as a service	30
	Defining Select Identity Resources	31
	To import Select Identity resources from a text file	31
	Replicating Select Identity Users in Select Access	34
	Manually Duplicating Accounts	34
	To create a duplicate entry in the Select Access directory server	34

To export the User ID attribute value in an HTTP header.....	35
Setting Policy on Specific Select Identity Resources	35
To set specific logout policy	35

1 Understanding HP OpenView Identity Management

With the implementation of HP OpenView Identity Management, all the tasks around user provisioning and authentication are streamlined.

Select Identity and Select Access form the backbone of HP's comprehensive Identity Management suite, delivering a full solution for complex identity management across the enterprise. This suite:

- Automates access control and user life-cycle management.
- Extends the enterprise through federation.
- Delegates management to business owners and the end users themselves.
- Allows processes to go online securely, without sacrificing ease of use.

Along with robust workflow, user self-service, reporting, and delegated administration capabilities, Select Access with Select Identity is the most comprehensive identity management system available. Together, they not only automate the process of user provisioning and account management, but also simplify your ability to secure user access to IT services and resources, including Select Identity's own functions.

Assumptions in this Document

This document assumes the following:

- That you have Select Identity installed and running on your network
 - That you understand the features and functions of Select Identity
 - That you thoroughly understand your application server and the implications of using a proxy with that server
 - That you have a working knowledge of Select Access and LDAP 3.0-compliant directory servers
 - That you have installed a supported application server for both Select Access and Select Identity: either BEA WebLogic and IBM WebSphere application servers
- ▶ The integration tasks described in this document are dependent on the application server you have chosen to implement.

Integrating the Select Access and Select Identity Frameworks

To understand the integration between Select Access and Select Identity, you need to understand the separate architectures of each and where critical integration points lie.

Understanding the Two Architectures

Select Identity identity management and provisioning architecture consists of the following components:

- A [Web Application server](#) that allows Select Identity resources to be deployed over the HTTP protocol. Select Identity relies on a Web Application server to serve the Select Identity interface pages, communicate with a database server to store and retrieve data, and send email based on an action performed through the Select Identity interface.
- A [relational database and server](#) that allow Select Identity to manage provisioned user data from a single source.

Select Identity's architecture must be coupled with the Select Access' architecture. Important components that are affected by this integration include:

- An [LDAP 3.0-compliant directory server](#) that allows you to create user accounts for Select Identity administrators. Information in the directory allows Select Access to manage authentication and authorization policies for explicit user/Select Identity resource combinations.
- An [Enforcer plugin](#) that intercepts all access requests on Select Identity's application server. The Enforcer plugin enforces the outcome of the access request that has been evaluated by the Policy Validator.
- **The Policy Builder:** the Java GUI that provides you with a view of all users (that is, Select Identity administrators and any other users your deployment requires) and available enterprise resources (for example, Select Identity functions). The combination is displayed as a hierarchical matrix which can be easily expanded and contracted to facilitate quick navigation.



For a complete overview of all Select Access components, refer to the *HP OpenView Select Access 6.2 Installation Guide*.

Understanding Critical Integration Points

To use Select Access to authenticate and authorize access for Select Identity resources, means that you have to configure Select Identity to use Select Access as its security provider. Integration primarily occurs through the following mechanisms:

- **An agent:** Depending on your application, the type of agent used to control access varies:
 - For IBM WebSphere servers: A proxy server, which acts as the host for the Enforcer plugin. The proxy is the access control point to Select Identity functions.
 - ⚠ If you are deploying Select Access and Select Identity using an IIS proxy, you must patch Select Access 6.0 with Patch 1. Otherwise, the unpatched version of the IIS Enforcer plugin does not perform personalization as required for this integration.
 - For BEA WebLogic: A custom security provider for WebLogic, which is distributed with Select Access installers. This security provider intercepts all resource requests and enforces the policy decisions made by the Policy Validator.
- **Personalization headers:** HTTP headers contain environment variables that allow Select Access and Select Identity to share user attributes used by each. User access results can be transferred from the Select Access directory to Select Identity by activating the personalization feature. That way, the User ID required by the Select Identity system is retained with the access result.

Example of an Integrated Network Architecture

Figure 1 shows a deployment of Select Access and Select Identity on the same corporate network. Notice how in most cases, the Select Access Enforcer plugin and the proxy/security provider are all installed on Select Identity's application server host computer. Other components for both Select Access and Select Identity can be deployed across the network as you require.

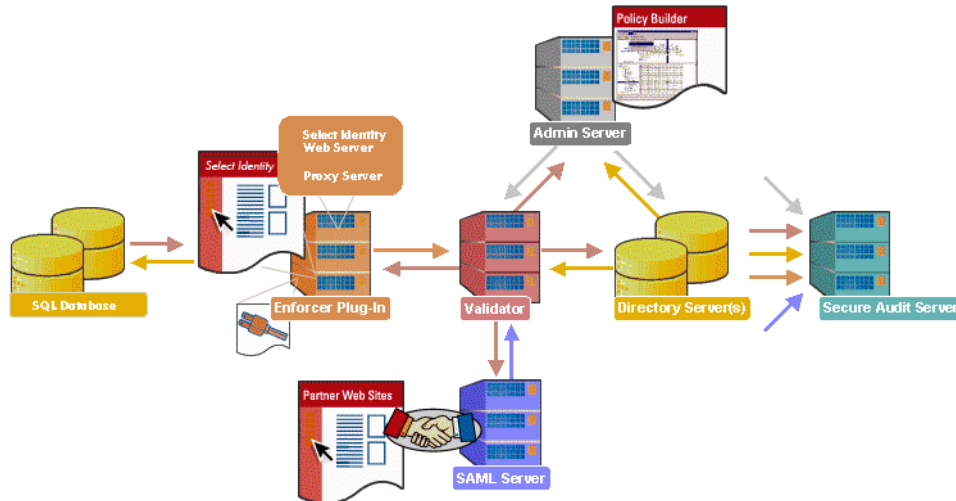


Figure 1 Select Access with Select Identity System Architecture

The Select Identity User Authentication Process

Select Access' authentication and authorization of Select Identity users takes place within a small number of basic steps. Select Access components communicate via XML documents known as queries and responses. XML offers Select Access complete flexibility for data transmission and integration into existing Select Identity functions and/or other enterprise resources and applications. Select Access' authentication and authorization process for Select Identity follows these steps:

- 1 A Select Identity administrator makes a request to access a Select Identity function to the web server.
- 2 The Select Access Enforcer plugin intercepts this request, and passes details of the request to the Select Access Policy Validator, including any authentication information the administrator has provided.
- 3 The Policy Validator searches for the administrator's user account and checks the policy data for the function requested. The Policy Validator caches the user and policy data from the directory for future retrieval.
- 4 Based on this combination of information, the Policy Validator returns a policy decision to the Enforcer plugin, including relevant personalization information, which at minimum contains the Select Identity User ID.
- 5 The Enforcer plugin enforces the validation result:
 - **If access is allowed:** The Enforcer plugin forwards the HTTP header that contains the User ID and any other personalization information to the proxy server. The proxy in turn forwards these results to the application server that loads the Select Identity function in the user's browser.

For each allow result, the application server records the User ID.

▶ WebLogic specifically records the ID in the principal object within the application server. This contrast with WebSphere's implementation, which requires that you configure Select Identity to retrieve the user data from the SSO token name you configure. For details, see [Step 4](#) on page 28 in the [To reconfigure Select Identity for WebLogic and WebSphere deployments](#) section.

- [If access is denied](#): The Enforcer plugin displays the access denied page.
 - [If more data is required](#): The Enforcer plugin displays the correct form to collect that data.
- 6 If the validation result returns an allow, Select Identity then retrieves the user ID from the application server.

Integration Task Overview

In order to integrate access management with identity management, this tandem deployment of the two HP OpenView products requires that you manually manipulate components, systematically modify properties, and actively enable features. You can separate these tasks into two distinct categories:

- Integration tasks that you must perform to get the two products to work together on the same network. These tasks are documented in [Chapter 2, Deploying the HP OpenView Identity Management Solution](#).
- Configuration tasks that you must perform post-integration to set up Select Access to protect Select Identity's resources. These tasks are documented in [Chapter 3, Protecting Select Identity Assets](#).

2 Deploying the HP OpenView Identity Management Solution

To ensure Select Access and Select Identity function properly as a unit, you must follow a specific series of steps to integrate them. [Table 1](#) summarizes the steps you need to perform, so that you configure and delegate its user authentication and authorization responsibilities to Select Access.

Table 1 Integration Task Overview

Integration Task	Details
1 Deploy all Select Access agents.	Deploying the Select Access Enforcer Plugin for BEA WebLogic on page 14 or Deploying IBM WebSphere with the IBM HTTP Proxy & the Select Access Enforcer Plugin on page 26
2 Reconfigure Select Identity to: <ul style="list-style-type: none">— Disable its internal log on mechanism. This is required so that Select Access acts as Select Identity's authentication and authorization agent.— Configure appropriate login/logout resources for this integration.	Reconfiguring Your Select Identity System on page 27

Deploying the Select Access Enforcer Plugin for BEA WebLogic

To ensure Select Access and the WebLogic server function properly as a unit, you must follow a specific series of steps to integrate them correctly. [Table 2](#) summarizes the steps you need to perform, to configure and delegate all authentication and authorization responsibilities to Select Access



If you are deploying against Select Access 6.2, due to requirements with Select Access' WebLogic Enforcer plugin, you must update your deployment of Select Access 6.1 with the most recent Engineering Patch. Please make sure you have installed the patch before starting any integration tasks.

Table 2 Integration Overview

Integration Tasks	Details
1 Install Select Access' WebLogic Enforcer plugin files on the same host as the WebLogic server.	Deploying Select Access on a BEA WebLogic Host on page 14
2 Define the classpath for Enforcer JAR files. The classes in these JAR files are used by the BEA WebLogic server.	Defining Select Access Classpaths on page 16
3 Create a properties file for the Enforcer plugin.	Storing WebLogic Connection Information on page 18
4 Configure the BEA WebLogic server to use Select Access as its Security Provider.	Making Select Access the WebLogic Security Provider on page 20
5 Set default security constraints for WebLogic resources so that they are protected as they are added to the Policy Builder.	Setting Default Security Constraints for WebLogic Resources on page 21
6 Implement Form-based authentication.	Enabling Form-based Authentication on WebLogic on page 22

Deploying Select Access on a BEA WebLogic Host


An important step in deploying Select Access is ensuring all the required files are successfully transferred to the BEA WebLogic host. These files include:

- specific Enforcer API files
- and `enforcer_bea.xml` bootstrap file. This file contains the configuration parameters that are required by the Enforcer API classes that are part of Select Access' WebLogic Enforcer plugin JAR files.

To get these required files on the BEA WebLogic host, simply install any of the recommended Enforcer plugins listed in the procedure below. To create the bootstrap file, refer to [To configure the plugin and create the XML bootstrap file](#) that follows.

To install an Enforcer plugin


- 1 Run the Select Access 6.2 installer.
- 2 Follow the instructions described in [Chapter 3, Installing Select Access](#), in the *HP OpenView Select Access 6.2 Installation Guide*.
- 3 When you have reached the **Choose HP OpenView Select Access Components** screen, select one of the following check boxes to install all of the correct Enforcer API files:
 - Sun/Netscape/iPlanet Enforcer plugin
 - IIS Enforcer plugin
 - Apache 2 Enforcer plugin

 You do not need to have web server software installed on WebLogic's host machine to install and configure an Enforcer plugin.
- 4 Click **Next** and finish the installation as required.

To configure the plugin and create the XML bootstrap file

- 1 Run the Setup Tool as described in [Chapter 4, Configuring Select Access](#), in the *HP OpenView Select Access 6.2 Installation Guide*.
 - 2 Click **Next** until you reach the **Generic Enforcer plugin** setup screen. This wizard creates the XML bootstrap file Select Access' WebLogic Enforcer plugin requires.
 - 3 Click **Browse** and select the folder to where the XML file will be created.
 - 4 Name and specify the path for the file you are about to generate. HP recommends that you save the file to the same folder as the Web Logic Domain. For example, `<WL_Domain_path>/enforcer_bea.xml`. This file name must match the path file name you configure in your properties file. For details, see [Sample bea_enforcer.properties file](#) on page 19.
 - 5 Enter your administration login credentials in the **Contact the Administration Server** setup screen and click **Next**.
 - 6 On the **General** setup screen, choose the **Custom** configuration option.
 - 7 Click **Next** on subsequent screens to accept default values provided by the wizard.
 - 8 When you reach **Ignore Filenames** setup screen, configure it to bypass authentication for security insensitive files and thereby enhance the Enforcer plugin's performance:
 - a Click **Add**.
 - b In separate rows, create the following resource list:

```
/lmz/images/*
/lmz/StyleSheet/*
/lmz/Javascript/*
/lmz/help/*
```

 This integration requires that you type path names exactly as shown.
- When you are finished, click **Next**.
- 9 Accept all defaults provided by the wizard until you reach the **Finish** setup screen.

- 10 Click **Finish** to commit your configuration to both the directory server and the Enforcer plugin's XML bootstrap file. At this point, the proxy server is ready to forward requests to Select Identity and all of the requests will now be reviewed by Select Access. You now need to define your security policy and configure Select Access to pass user information to Select Identity. For details, see [Chapter 3, Protecting Select Identity Assets](#).

Defining Select Access Classpaths

Because the WebLogic server uses the Enforcer API classes in Select Access' WebLogic Enforcer plugin JAR files, you need to append the required Select Access files to the CLASSPATH of WebLogic server.

Typically, you define CLASSPATHs in the WebLogic server's startup script. By default the name of startup script is `startWebLogic.cmd`, which is stored in the WebLogic Server Domain directory.

You need to append the following JAR files to the CLASSPATH section of the WebLogic server's startup script:

- `SAPrincipal.jar`: Contains the `SAPrincipalImpl` class that extends the `WLSUser` interface. It allows WebLogic to use its default security providers with Select Access' Enforcer plugin in the same security realm. The `SAPrincipalImpl` class accomplishes this by representing Select Access users as native users in the WebLogic system.
- `EnforcerAPI.jar`: Contains all the classes used to create an Enforcer plugin. These files make use of the following JAR files:
 - `AcmeCrypto.jar`
 - `bcprov-jdk14.jar`
 - `castor-0.93.19-xm.jar`
 - `jdom.jar`
 - `ldapjdk.jar`
 - `protomatter.jar`
 - `SAResourceBundle.jar`
 - `shared.jar`
 - `xerces.jar`
 - `xml.jar`

To append Enforcer API JAR files to the classpath

- 1 Create a new folder for holding all the JAR files Select Access' WebLogic Enforcer plugin requires:

```
<WebLogic_install_path>/server/lib/sa
```


- Copy the JAR files listed in [Table 3](#) into the folder you just created.


Table 3 Files You Need to Manually Install

Required File	Where to Get It
SAPrincipal.jar	/solutions/WebLogic folder on the Select Access 6.2 product CDs or the most recent engineering patch for 6.0
bcprov-jdk14.jar SAResourceBundle.jar jdom.jar protomatter.jar xerces.jar castor-0.9.3.19-xml.jar ldapjdk.jar shared.jar xml.jar	<SA_install_path>/shared/ jetty/policy_builder/ protected
EnforcerAPI.jar AcmeCrypto.jar	<SA_install_path>/shared

- Open your startup script for the WebLogic server. By default the name of startup script is `startWebLogic.cmd`, which is stored in the domain directory you configured with the WebLogic Configuration wizard.
- Create an `SA_LIB` parameter in the startup script that points to the Select Access folder you created in [Step 1](#). Using the example provided in that step, the variable added to the startup script is:

```
set SA_LIB=C:\bea\weblogic81\server\lib\sa
```

- Create an `SA_CLASSPATH` system environment variable that includes all the JAR files listed in [Table 3](#). This defines the correct classpaths to the required Select Access JAR files described in [Table 3](#). Without these paths, the WebLogic server would not know where to find the files for the Enforcer plugin. [Sample of the SA_CLASSPATH variable](#) on page 17 shows the contents of this new variable.

 The paths defined in the sample, use the `SA_LIB` variable you created in [Step 4](#).

- Locate the `CLASSPATH` variable and append `SA_CLASSPATH` variable as another value for the `CLASSPATH` variable. For example:

```
set CLASSPATH=%CLASSPATH%;%SA_CLASSPATH%
```

- Save the changes to this file.

Sample of the SA_CLASSPATH variable

The following is a sample of the `SA_CLASSPATH` variable:

```
set SA_CLASSPATH=%SA_LIB%;%SA_LIB%\bcprov-jdk14.jar;  
%SA_LIB%\AcmeCrypto.jar;
```

```
%SA_LIB%\SAPrincipal.jar;%SA_LIB%\castor-0.9.3.19-xml.jar;
%SA_LIB%\EnforcerAPI.jar;%SA_LIB%\jdom.jar;
%SA_LIB%\ldapjdk.jar;%SA_LIB%\protomatter.jar;
%SA_LIB%\SAResourceBundle.jar;%SA_LIB%\shared.jar;%SA_LIB%\xerces.jar;
%SA_LIB%\xml.jar
```

Storing WebLogic Connection Information

In order for the Enforcer plugin to send valid queries to the Policy Validator for WebLogic resources, it must know the service name and root resource path defined for the protected WebLogic server. The Enforcer plugin reads this information from a configuration file called `bea_enforcer.properties`.

This file is not native to Select Access nor to the WebLogic server. In order to use Select Access with WebLogic, you must create this configuration file and save it to the domain directory you configured with the WebLogic Configuration wizard.



The service name you define as a parameter in this file must exactly match the host name and port that you configured for your WebLogic services. You will eventually add these services to the Policy Matrix. See [Chapter 3, Protecting Select Identity Assets](#) for information about adding WebLogic services with the Policy Builder.

To create the `bea_enforcer.properties` file

- 1 In a text editor create a new file.
- 2 Add the parameters listed in [Table 4](#) to the file.
- 3 Save the file. When you are finished, your properties file should look similar to the one shown in [Sample `bea_enforcer.properties` file](#) on page 19.

Table 4 Properties You Need to Add

Property Name	Description
EnforcerAPIConfigFile	This file name must match the file name you defined for your XML configuration file. For details, see To configure the plugin and create the XML bootstrap file on page 15.
Service	Identifies where the WebLogic server is located using the host name for the computer on which the WebLogic server is located and the a port number. This parameter uses the following syntax: Service=<hostname>:<port>

Table 4 Properties You Need to Add (cont'd)

Property Name	Description
Resource	<p>Identifies the root directory that all queries to the Policy Validator start with. This parameter uses the following syntax:</p> <p>Resource="<root_path>"</p> <p>Note: Typical deployments often use a value of "/" as the root path.</p>
LogLevel	<p>Because audit configuration has not been enabled for this implementation of Select Access' WebLogic Enforcer plugin, you may want to include this parameter if you want to collect log messages generated by the Enforcer plugin. HP recommends you include this property to minimize output to platform-specific locations. You can choose any of the following values, listed in order of severity:</p> <ul style="list-style-type: none"> • fatal • error • warning (the default setting) • info • debug <p>These values correspond to Enforcer plugin options you can configure with Select Access' Setup Tool. All log messages are forwarded to the location you defined in the Enforcer plugin's configuration file. For details on these values see Configuring Enforcer-Specific Audit Settings on page 146 in the <i>HP OpenView Select Access 6.2 Installation Guide</i>.</p>
SecurityRealm	<p>The security realm used by Perimeter Authentication. Set it to the Select Access' Security Realm. For details, see Making Select Access the WebLogic Security Provider on page 20.</p>

Sample bea_enforcer.properties file

The following is a sample of the `enforcer.properties` file:

```

EnforcerAPIConfigFile=enforcer_bea.xml
Service=myBEAas:7001
Resource=/
LogLevel=info
SecurityRealm=SARealm

```

Making Select Access the WebLogic Security Provider

A critical integration step is to make Select Access the Security Provider for the WebLogic server. This step ensures that Select Access components, not WebLogic's native mechanisms, protect the application server and its available resources. Setting up Select Access as the Security Provider requires that you follow the steps described in [Table 5](#).

Table 5 Making Select Access the Security Provider

Setup Task	Configuration Details
<p>Step 1: Copy the Java archive file.</p> <ul style="list-style-type: none"> For 6.2 releases, check the Select Access product CDs in the <code>solutions\WebLogic</code> folder. For 6.1 releases, check the contents of available Select Access engineering patches. 	<ol style="list-style-type: none"> Stop the WebLogic server. Copy <code>SASecurityProviders.jar</code>. Move it to the following folder: <code><WebLogic_install_path>/server/lib/mbeatypes</code> Restart the server.
<p>Step 2: Set up the WebLogic server to use Select Access as its Security Provider.</p> <p>Note: For specific procedural details for any of the tasks listed for this step, refer to the WebLogic server's documentation.</p>	<ol style="list-style-type: none"> Open the WebLogic Server Console in a web browser of your choice. Create and configure a new security realm for Select Access by using default settings provided in the Create a new Realm page. The only unique property you want to manually configure is the realm name: for example <code>SArealm</code>. Note: This name must match the parameter for the realm in the <code>bea_enforcer.properties</code> file. Click the realm you created in Step 2, and create and configure its Security Providers. You must set up these according to HP's recommendations. For details, see To set up Select Access as a WebLogic Security Provider that follows. Because you can have multiple security realms in a WebLogic server domain, but only one set as the default (active) realm, ensure the Select Access realm you created in Step 2 is the default security realm of your domain. To activate the new security provider, stop and restart the WebLogic server.

To set up Select Access as a WebLogic Security Provider

- In the `<realm_name>` page for the realm you created in Step 2 of [Table 5](#), click the **Providers** tab.
- In the **Adjudicators** tab, create and configure a `Default Adjudicator`. Use defaults provided by WebLogic, except in the **Details** tab: ensure that you uncheck the **Require Unanimous Permit** box.
- In the **Auditors** tab, optionally create and configure a `Default Auditor`. Choose the desired severity, according to your security requirements.

- 4 In the **Authentication** tab, create and configure the following Authentication Providers in the order outlined in [Table 6](#).

Table 6 Required Order of Authentication Providers

Provider Name	Configuration Details
1 SAAuthenticator	In the General tab, set the Control Flag set to OPTIONAL .
2 Default Authenticator	In the General tab, set the Control Flag set to OPTIONAL .
3 SAIdentity Asserter	In the General tab, make <code>PolicyUser</code> an active token by moving it from the Available column to the Chosen column. In the Details tab, uncheck the Base64Decoding Required option.
4 Default Asserter	In the General tab, make <code>AuthenticatedUser</code> an active token by moving it from the Available column to the Chosen column.

- 5 In the **Authorizers** tab, create and configure `SAAuthorizer` and `Default Authorizer`, by using WebLogic's default values.
- 6 In the **Credential Mappers** tab, create and configure a `Default Credential Mapper`, by using WebLogic's default values.
- 7 In the **Role Mappers** tab, create and configure a `Default Role Mapper`, by using WebLogic's default values.

Setting Default Security Constraints for WebLogic Resources

Due to the design of the J2EE architecture, WebLogic resources cannot use Select Access for authentication by default. While you can still enable authentication in the Policy Builder for each resource, the authentication mechanisms will have no effect. This restriction exists because security for each J2EE resource is controlled by a deployment descriptor XML file that is resource-specific. However, this descriptor file does not initially include default security constraints.

To enable Select Access authentication, you must manually modify the WebLogic server Startup Scripts. This modification allows you to configure the WebLogic server to protect all WebLogic resources by default. Select Access then passes a flag to the WebLogic server upon the application server's startup. This flag forces it to protect all resources, irrespective of any security constraint information specified in the deployment descriptors.



Users can be transferred from the WebLogic internal user store to an LDAP directory using the Migrate functionality in the WebLogic Admin Console.

To configure the WebLogic server to protect all resources

- 1 Open the `<WLDomain_path>/startWebLogic.cmd` file.
- 2 Do one of the following:

- If you have a `JAVA_OPTIONS` variable, append the following parameter to the end of it:

```
-Dweblogic.security.fullyDelegateAuthorization=true
```

- If you do not have a `JAVA_OPTIONS` variable, create this variable and set the required parameter for it:

```
set JAVA_OPTIONS=  
-Dweblogic.security.fullyDelegateAuthorization=true
```

- 3 Ensure the `JAVA_OPTIONS` variable appears in the line that starts the WebLogic server. This line is automatically included if you used the Configuration wizard to setup WebLogic. Otherwise, if you have manually created your file, you may need to append this line.

[Sample startup line](#) on page 22 shows this variable in bold. All other variables vary depending on your deployment of WebLogic.

- 4 Save the file. By default, all WebLogic resources will be protected by Select Access with these changes once you restart the server after reconfiguring Select Identity.

▶ You must set the corresponding access policy for each user in the Policy Builder. See [Chapter 3, Protecting Select Identity Assets](#) for details.

Sample startup line

The example below shows how to use the `JAVA_OPTIONS` variable in the startup line:

```
%JAVA_HOME%\bin\java %JAVA_VM% %MEM_ARGS% %JAVA_OPTIONS%  
-Dweblogic.Name=%SERVER_NAME% -Dweblogic.management.username=%WLS_USER%  
-Dweblogic.management.password=%WLS_PW%  
-Dweblogic.ProductionModeEnabled=%PRODUCTION_MODE%  
-Djava.security.policy="%WL_HOME%\server\lib\weblogic.policy"  
weblogic.Server
```

Enabling Form-based Authentication on WebLogic

When integrating Select Access and Select Identity, to make the user experience more effective and seamless you want users to only authenticate once with Select Access but still have access to Select Identity resources. To accommodate this goal, you need to configure both products to use Form-based authentication. The form you use can either be:

- The login form template provided by HP.
- Your own custom login page.

▶ Login pages should typically not include any protected resources. Otherwise, users may not be redirected back to the original page after a successful login. For details, see [Using Graphics on Login Pages](#) on page 25.

To setup Form-based authentication with multi-domain SSO, you need to complete the tasks described in [Table 7](#):

Table 7 Configuration Tasks for Form-based Authentication

Task Required	Details
1 Stop the web server.	
2 In Select Identity, define the Select Access login servlet as your web application's login page. You can locate the parameter in Select Identity's deployment descriptor file called <code>web.xml</code> .	To set up SSO in Select Identity on page 23
3 In Select Access, configure an Authentication server and corresponding login form in the Policy Builder.	Chapter 7 in the <i>HP OpenView Select Access 6.2 Policy Builder Guide</i>
4 In Select Access, run the Setup Tool and configure SSO for each Enforcer plugin for each third-party web server and WebLogic server.	Chapter 8 in the <i>HP OpenView Select Access 6.2 Installation Guide</i>
5 Restart the web server.	

To set up SSO in Select Identity

- 1 Extract the `lmz.war` file from `<SI_path>/lmz.ear`.
- 2 Extract the `web.xml` file from the `lmz.war` file you just extracted. Ensure you preserve the file's full path.
- 3 Open the `web.xml` file you just extracted in a text editor.
 - a Add and/or configure the tags listed in [Table 8](#) to this file.
 - b Save the changes to this file.

Table 8 web.xml Tags Required

Tag	Sub-tag	Description
servlet	servlet-name	Sets name to any string. This string must meet the application's requirements.
	servlet-class	Sets the class of the login servlet. The class must be fully-qualified.
	run-as	Sets the role-name to a WebLogic administrator (with the corresponding role) defined by the default Authentication Provider. The administration role allows Select Access' servlet to perform Perimeter Authentication. Tip: When you created Select Access' security realm, all default user, group, role and policies are added to it. This includes a default user that Select Access uses to boot the WebLogic server. You can use this name as the run-as role name.
servlet-mapping	servlet-name	This string must: <ul style="list-style-type: none"> • Meet the application's requirements. • Must match the servlet-name in the servlet tag.
	url-pattern	Maps the servlet to a url string pattern. This string must meet the application's requirements.
login-config	auth-method	Sets the method of authentication. Define this method as FORM.
	realm-name	Sets a specific realm. If you do not supply a realm name, the default realm is used.
	form-login-config	Sets: <ul style="list-style-type: none"> • form-login-page to the same URL pattern as in servlet-mapping. • form-error-page to the application's login error page. Leave the value of this tag empty; Select Access does not use this value.

See [Sample web.xml file](#) for an example of a completed web.xml file.

- 4 Package the modified web.xml file into the lmz.war file. Ensure that the file's full path is preserved.
- 5 Package the lmz.war file into the lmz.ear file.

Sample web.xml file

The following is an example of the additions to the `web.xml` file. Place the different sections within the correct context of your own file so proper order is maintained.

```
<web-app>
  <!-- Define the Servlets within the Web Application -->
  <ervlet>
    <ervlet-name>LoginServlet</ervlet-name>

    <ervlet-class>com.hp.ov.selectaccess.solutions.weblogic.securityprovide
r.LoginServlet</ervlet-class>
    <run-as>
      <role-name>weblogic</role-name>
    </run-as>
  </ervlet>
  <!-- Define Servlet mappings to urls -->
  <ervlet-mapping>
    <ervlet-name>LoginServlet</ervlet-name>
    <url-pattern>login.htm</url-pattern>
  </ervlet-mapping>
  <login-config>
    <auth-method>FORM</auth-method>
    <realm-name>SAR realm</realm-name>
    <form-login-config>
      <form-login-page>/login.htm</form-login-page>
      <form-error-page></form-error-page>
    </form-login-config>
  </login-config>
</web-app>
```

Using Graphics on Login Pages

If your login page has some image files in it, check that those image files are not protected by any Select Access access policies. You can use one of two methods:

- Setting up access in the Policy Builder: Set an Allow access policy for all images on the login page for all unknown users.
- Setting up Select Access' WebLogic Enforcer plugin to ignore all image files with the **Ignored Filenames** setup screen when creating your Enforcer configuration XML file. Make sure you check the **Ignore GIF images** and the **Ignore JPEG images** boxes. For details, see [Chapter 8, Configuring the Enforcer Plugins](#) in the *HP OpenView Select Access 6.2 Installation Guide*.

Deploying IBM WebSphere with the IBM HTTP Proxy & the Select Access Enforcer Plugin

The IBM HTTP server is built into the WebSphere application server, which consequently is the only recommended proxy server for WebSphere deployments. Nonetheless, like other Apache web server deployments, Select Access controls access to Select Identity with an Apache 2 Enforcer plugin agent that you install on the WebSphere server. Table 9 lists the steps to successfully integrate this plugin with an IBM HTTP server proxy.

Table 9 Tasks Required to Deploy Select Access Agents

Deployment Task	Details
1 Run WebSphere's administration tool to append an entry for a module filter in the <code>httpd.conf</code> file. This entry allows the HTTP server to connect to WebSphere.	WebSphere Server Documentation
2 Disable the fast response cache. This prevents unauthenticated users from accessing pages from the cache.	To disable fast response caching on page 26
3 Install and specifically configure the Apache 2 Enforcer plugin for a Select Identity integration.	To deploy the IBM HTTP Enforcer plugin on page 26

To disable fast response caching

- 1 With the port you configured when you installed WebSphere, login to the IBM HTTP Administration server.
- 2 In the **Performance** properties for the IBM Administration server, click **Server Settings**.
- 3 Set the **Enable Fast Response Caching** to **No**.

To deploy the IBM HTTP Enforcer plugin

- 1 Install the IBM HTTP Enforcer plugin. For details, refer to the *HP OpenView Select Access 6.2 Installation Guide*.
- 2 When you have installed the Enforcer plugin, the Setup Tool appears. Click **Next** until you reach the Setup wizard for the IBM HTTP Enforcer plugin and then click **Configure**. With this deployment, you need to configure the plugin to tune its performance when controlling access to Select Identity resources.
- 3 Enter your administration login credentials in the **Contact the Administration Server** setup screen and click **Next**.
- 4 On the **General** setup screen, choose the **Custom** configuration option.
- 5 Click **Next** on subsequent screens to accept default values provided by the wizard.
- 6 When you reach **Ignore Filenames** setup screen, configure it to bypass authentication for security insensitive files and thereby enhance the Enforcer plugin's performance:
 - a Click **Add**.

- b In the new row that appears, list resources that also do not require explicit authentication.

For example, you should, at minimum, build a list that includes the following resources:

```
/lmz/images/*  
/lmz/StyleSheet/*  
/lmz/Javascript/*  
/lmz/help/*
```



This integration requires that you type path names exactly as shown.

- c When you are finished, click **Next**.
- 7 Accept all defaults provided by the wizard until you reach the **Finish** setup screen. Configure it as follows:
- a Select the following check boxes:
 - **Update web server configuration to load the Enforcer Plugin.** This allows the Setup Tool to append an entry for the Enforcer module in the server's configuration file.
 - **Restart web server.** This restarts your the server after so it uses the newly modified configuration file.
 - b Click **Finish** to commit your configuration to both the directory server and the Enforcer plugin's XML bootstrap file. At this point, the proxy server is ready to forward requests to Select Identity and all of the requests will now be reviewed by Select Access. You now need to define your security policy and configure Select Access to pass user information to Select Identity. For details, see [Chapter 3, Protecting Select Identity Assets](#).

Reconfiguring Your Select Identity System

Select Identity includes its own log on mechanism. This mechanism, left enabled, would conflict with Select Access' authentication features. Therefore, you must disable Select Identity-based authentication and reconfigure login as described below.

- [The login page](#) is required to give users that are not allowed access to Select Identity functions an alternate network location and/or starting point. A suitable resource for this configuration parameter may be a home page or a portal page. A suitable resource for this configuration parameter may be a message page that notifies the user that only administrators are allowed access to Select Identity resources.
- [The logout page](#) is required to give users that log out of Select Identity an alternate network location. These users may log out of Select Identity but still remained logged in with Select Access. A suitable resource for this configuration parameter may be a home page or a portal page.



To log out of Select Access entirely (and thereby delete the cookie that gives the user access to other Select Access-protected corporate resources), you need to configure a Conditional Rule that includes a logout terminal point and apply it to the user/resource combination in question. For details, see [Setting Policy on Specific Select Identity Resources](#) on page 35.

To reconfigure Select Identity for WebLogic and WebSphere deployments

- 1 Stop Select Identity.
- 2 Open the `<si_install_path>\sysArchive\TruAccess.properties` file.
- 3 For all deployments, modify the following configuration parameters as follows:

- `truaccess.authentication=off`
- `truaccess.loginURL=<MyInWebPage>`

Where `<MyInWebPage>` is the address of a web resource that Select Identity displays when a valid network user who is not a known/registered user in Select Identity tries to log in.

For example:

```
loginURL=http://localhost/support/noSIaccess.htm
```

- `truaccess.logoutPage=<MyOutWebPage>`

Where `<MyOutWebPage>` is address of a web resource that Select Identity displays when a user logs out of Select Identity.

For example:

```
truaccess.logoutPage=http://localhost/user_portall/portal.asp
```

- 4 For all deployments, modify the following configuration parameter as follows:

```
truaccess.sso.token.name=HTTP_SA<my_header_name>
```

Where `<my_header_name>` is the variable name you wish Select Access to export user data to using its personalization function. Values in Select Access and Select Identity must match exactly. For details on how to configure this HTTP header variable in Select Access, see [To export the User ID attribute value in an HTTP header](#) on page 35.

For example, if you want to export user data in the `UID` variable, the SSO token name you would define is `HTTP_SAUID`.

- 5 Restart the Select Identity application to implement these changes.

- ▶ Users cannot access Select Identity applications until the proxy server and Select Access have been completely deployed on the network.

3 Protecting Select Identity Assets

Securing Select Identity resources with Select Access requires that you register its resources using the Policy Matrix, the grid-like interface that is the main administrative element in the Policy Builder. [Table 10](#) lists the tasks required to Select Access-protect your Select Identity resources.

Table 10 Securing Resources in the Policy Builder

Task	Details
1 Register security sensitive resources with Select Access by adding them to the Resources Tree in the Policy Matrix.	Registering Select Identity Resources on page 29
2 Replicate Select Identity administrator accounts on Select Access so a user entry exists for them in Select Access' directory server.	Replicating Select Identity Users in Select Access on page 34
3 Assign access and authentication policies to your user and Select Identity resource combinations.	Setting Policy on Specific Select Identity Resources on page 35

Registering Select Identity Resources

Because of the way the resources it uses are built into the product, Select Identity does not currently support an automated network discovery of its resources. Therefore, the registration of Select Identity resources you want to Select Access-protect is a manual process that requires you to:

- Define the services to which the resources belong. For details, see [Defining the Select Identity Services](#) on page 29.
- Add individual Resources Tree entries to the service's branch. For details, see [Defining Select Identity Resources](#) on page 31.

Defining the Select Identity Services

Services are organized by protocol in the Resources Tree with each protocol/server combination being treated as a separate service. This helps you to easily identify and locate the resources you want to set access policies against. [Figure 2](#) shows a portion of the Resource Access portion of the Resources Tree to illustrate how resources might be organized on a WebLogic application server.

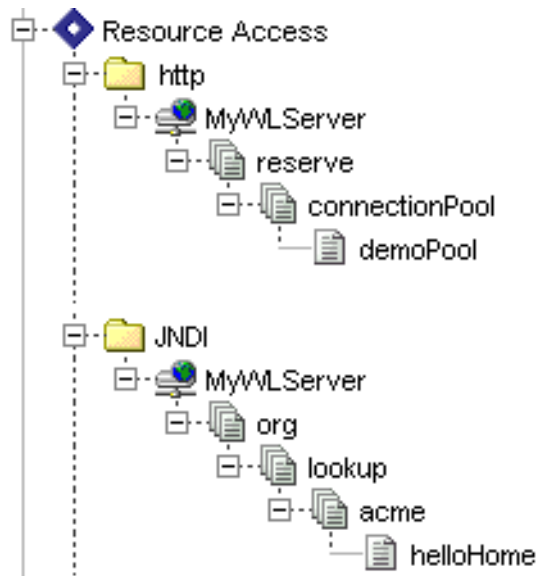



Figure 2 Sample Hierarchy for Resource Access Assets

To add Select Identity's application server as a service

- 1 If you do not already have one, create a folder named HTTP in Resource Access section. Do this by:
 - a Right-clicking Resource Access branch and clicking **New** → **Folder**. The **New Folder** dialog appears.
 - b Typing HTTP as the folder name.

This allows you to organize Select Identity resources by the HTTP protocol.
- 2 Create the Select Identity service entry. Do this by:
 - a Right-clicking on the HTTP protocol folder and clicking **New** → **Service**. The **New Service** dialog appears.
 - b In the **Name** field, typing a name for the Select Identity service entry.
 - c Configuring the proxy server's contact information:
 - Click the **Add** button to create a new server definition.
 - Select **HTTP** as the protocol.
 - Type the hostname/IP address and port number for the Select Identity proxy and port number.

 Do not configure the contact information for the application server.
- 3 If you are deploying Select Identity with WebLogic:
 - a Repeat the first two steps to create service entries for the following protocols:
 - EJB
 - JNDI
 - JDBC

- WebLogic
- ▶ The WebLogic protocol is used by WebLogic when accessing system resources like the Admin Console.
- b Set an Allow policy on the **Unknown Identities** column for the first three of these resources as they will be used by Select Identity.
- c Protect the service entry for the WebLogic protocol with the same authentication service used for the HTTP/HTTPS protocol. Then ensure you wet an Allow police for Known Identities only.

Defining Select Identity Resources

In order to restrict access to Select Identity functions, you must register functions as resources with Select Access before they can be protected. Since Select Identity does not currently allow Select Access to automatically discover these functions, you must define them manually, by using either of the following techniques:

- Importing a resource list from a text file. HP recommends you register resources with this option, because it less error-prone than adding manually via the Policy Builder. For details on importing a resource list, see [To import Select Identity resources from a text file](#) on page 31.
- Adding each resource to the service manually via the Policy Builder's **New Resource** dialog box. This option is more time consuming and more error prone than importing a list. HP recommends that you only use this option for smaller maintenance tasks (that is, adding one or two resources to your site). For details, refer to the *HP OpenView Select Access 6.2 Policy Builder Guide*.

To import Select Identity resources from a text file

- 1 List the Select Identity resources in a text file. [Table 11](#) on page 32 lists the resources you need to enumerate in this file. To reuse this list:
 - a Cut and paste the URLs into a text file.
 - b Substitute the place holder with your server's name or IP address and port number.
 - c Save the file.
- 2 Right-click on the Select Identity service that you created, and click **Run Discovery** → **Resources**. The **Discover Network Resources** dialog appears.
- 3 Select **Import Resource List From File** then click **Browse** to locate the file you created in [Step 1](#).

- 4 Click **Ok**. The icon at the bottom of the Policy Builder window flashes to indicate that the resource list is being imported successfully.

Table 11 URLs for Text File

Function	URL
Workflow Approvals	http://<server_name>:<port>/lmz/approval/search.do
Manage Attributes	http://<server_name>:<port>/lmz/attribute/search.do
Auto Discovery	http://<server_name>:<port>/lmz/userDiscovery/search.do
Import Configurations	http://<server_name>:<port>/lmz/config/importInit.do
Export Configurations	http://<server_name>:<port>/lmz/config/exportSearch.do
Manage Connectors	http://<server_name>:<port>/lmz/resource/manageConnectorList.do
Email Notifications	http://<server_name>:<port>/lmz/notification/search.do
External Calls	http://<server_name>:<port>/lmz/ExternalCall/init.do
Home Page	http://<server_name>:<port>/lmz/landing.do
Logout	http://<server_name>:<port>/lmz/logoff.do
Challenge/Response	http://<server_name>:<port>/lmz/policymgmt/init.do
Reconciliation Job	http://<server_name>:<port>/lmz/reconciliationjoblist/search.do
Reconciliation Task	http://<server_name>:<port>/lmz/reconciliationtaskstatus/search.do
Audit Reports	http://<server_name>:<port>/lmz/reports/audit/savedconfiglist.do
Manage Scheduled Audit Reports	http://<server_name>:<port>/lmz/reports/audit/savedconfiglist.do
Manage Audit Report Configurations	http://<server_name>:<port>/lmz/reports/audit/savedconfiglist.do
Manage Connectors	http://<server_name>:<port>/lmz/resource/manageConnectorList.do
Service Audit Report	http://<server_name>:<port>/lmz/control/reporting/audit/servicereportconfig

Table 11 URLs for Text File (cont'd)

Function	URL
User Audit Summary Report	http://<server_name>:<port>/lmz/control/reporting/audit/summary/userreportconfig
User Audit Report	http://<server_name>:<port>/lmz/control/reporting/audit/userreportconfig
Configuration Reports	http://<server_name>:<port>/lmz/reports/config/savedconfiglist.do
Manage Scheduled Configuration Reports	http://<server_name>:<port>/lmz/reports/config/savedconfiglist.do
User Configuration Detail Report	http://<server_name>:<port>/lmz/control/reporting/configuration/detail/userreportconfig
Manage Configuration Report Configurations	http://<server_name>:<port>/lmz/reports/config/savedconfiglist.do
User Configuration Summary Report	http://<server_name>:<port>/lmz/control/reporting/configuration/summary/userreportconfig
User Configuration Report	http://<server_name>:<port>/lmz
Request Status	http://<server_name>:<port>/lmz/requeststatus/search.do
Manage Resources	http://<server_name>:<port>/lmz/resource/search.do
Manage Rules	http://<server_name>:<port>/lmz/rule/search.do
Self Service	http://<server_name>:<port>/lmz/selfservice/getProfile.do?reload=true
Change User Context	http://<server_name>:<port>/lmz
Admin Roles	http://<server_name>:<port>/lmz/adminrole/search.do
Services	http://<server_name>:<port>/lmz/service/search.do
Create Business Relationship	http://<server_name>:<port>/lmz/control/taservicemgmt/createbizrelation

Table 11 URLs for Text File (cont'd)

Function	URL
Modify Business Relationship	http://<server_name>:<port>/lmz/control/taservicemgmt/modifybizrelation
Workflow Studio	http://<server_name>:<port>/lmz/workflowstudio/search.do
Manage Users	http://<server_name>:<port>/lmz/user/search.do

Replicating Select Identity Users in Select Access


Select Access currently requires that user accounts be stored in an LDAP 3.0-compliant directory server. Select Access components look up policy data stored for the user in this directory to authenticate and authorize user access. Therefore, at minimum, you must replicate User IDs in the Select Access system for individuals who perform administrative roles in Select Identity. However, if you want Select Access to control other corporate resources, you can replicate accounts for all known users with any other attributes your specific deployment requires.

Manually Duplicating Accounts

With the current releases of the two products, this integration requires that you manually create a new user entry for each Select Identity user you want Select Access manage access for. You do not need to populate other elements of the directory. User IDs between both systems do not need to even match, although exact matching does simplify things, because Select Access exports this ID to Select Identity in an HTTP header that is typically used by Select Access for personalization purposes.

Depending on the directory server you are using to support Select Access, the User ID is typically be stored in any of the following attributes: UID, SN, and CN. However, you are not restricted in which attribute you use; you just need to remember to export the corresponding attribute using Select Access' personalization feature. You then also need to ensure that the variable you use matches the value you configured in the `truaccess.properties` file.

To create a duplicate entry in the Select Access directory server

- 1 If you have not already done so, configure your user data location. You may have already set up this location when you installed Select Access. However, if you did not, you can use the Policy Builder to configure this directory location now. For details, see [To add or modify an identity location](#) on page 30 in the *HP OpenView Select Access 6.2 Policy Builder Guide*.
- 2 Use the Policy Builder to create Select Identity users.
 -  If you are using Select Access to protect other corporate resources in addition to Select Identity resources, you may want to organize users logically using folders, groups, or roles.

To do this:

- a Right-click a folder or user location branch in the Identities Tree and click **New** → **User**. The **New User** dialog appears.
- b Configure the following directory attribute properties:
 - **Last Name**: Required by Select Access components.
 - **Common Name**: Required by Select Access components.
 - **User ID**: Required by Select Access and Select Identity components.

All other properties are optional. You may configure any other attributes if your deployment requires it.

To export the User ID attribute value in an HTTP header

- 1 Right-click the square in the SelectAuth column beside the entry, then click **Enable SelectAuth**. The **SelectAuth Properties** dialog box appears.
- 2 Click the **Personalization** tab.
- 3 Click the **User Data** subtab and do the following:
 - a Select the **Store user attributes in** check box.
 - b Click **Add** to create a new definition.
 - c In the first column, type the attribute name you wish to export.
 - d In the second column, type the corresponding **Environment Variable Name** that Select Access will export the attribute to. Depending on your Enforcer plugin type, the plugin will prepend a Select Access-specific prefix to it. For details on how to configure Select Identity with this information, see [To reconfigure Select Identity for WebLogic and WebSphere deployments](#) on page 28.

Setting Policy on Specific Select Identity Resources

Select Access can handle all authorization requirements you have, when you use any combination of allow/deny/conditional policies. You can set policy as your sensitivity of resources require. Using built-in inheritance rules, the Policy Builder allows you to set policy for all your known users quickly and easily. For an overview, refer to the *HP OpenView Select Access 6.2 Policy Builder Guide*.

However, you may want to assign unique policies to the login and logout resources you configured in [To reconfigure Select Identity for WebLogic and WebSphere deployments](#) on page 28. You may also require a logout conditional rule that also logs the user out of Select Access and thereby destroys the cookie that gives them access to other network resources, not just Select Identity functions. For details, see the section that follows.

To set specific logout policy

- 1 Where the **Known Users** column and the Select Identity resource you configured for the login page parameter intersect, right-click the cell and click **Allow Access**.
- 2 Where the **Known Users** column and the Select Identity resource you configured for the logout page parameter intersect, right-click the cell and click **Allow Access**.

- 3 If you require that users also logout of Select Access to protect sensitive resources, you may want to create a Conditional Rule. Where the **Known Users** column and the Select Access logout intersect, right-click the cell and click **Conditional Rule**.
 - If you have already created a rule with a standalone logout terminal point in it, choose this rule from the list now.
 - Otherwise, create a rule that simply includes a logout terminal point.