HP OpenView Select Access

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

Software Version: 6.2

Network Integration Guide

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

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 developed by The Cryptix Foundation Limited.
- Cryptographic software written by Eric Young.
- 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.

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 Introduction

Identity management touches upon almost every aspect of the HP Adaptive Enterprise vision, affecting access to information across hardware, software, network resources, application servers, enterprise applications, and web portals within an organization and across organizations via business-to-business transactions.

For managing an adaptive enterprise (one that can respond quickly to change) HP OpenView Select Access provides systematic and secure user access to third-party network services and the enterprise resources they deploy. Select Access provides integrated infrastructure management that drastically reduces corporate IT costs.

By using a highly scaleable and extensible architecture, Select Access integrates with most dynamic IT environments that include:

- Wireless, web, non-web and legacy applications support
- Native LDAP v3 directory serves as the repository for user, resource and policy data; it works with existing user data and directory schema
- Leading web J2EE-compliant application and portal servers
- Popular authentication schemes to allow flexibility in strength of user identification

Audience

This document is intended for administrators of HP OpenView Operations deploying Select Access as their Identity Management solution with any of the third-party technologies Select Access supports. This guide enables these administrators to successfully integrate Select Access with their legacy systems.

The Select Access Documentation Set

This manual refers to the following Select Access documents. These documents are installed with Select Access and are available in the `<install_path>/docs` folder.

- *HP OpenView Select Access 6.2 Installation Guide*, © Copyright 2000-2006 Hewlett-Packard Development Company, L.P. (`installation_guide.pdf`)
- *HP OpenView Select Access 6.2 Policy Builder Guide*, Copyright 2000-2006 Hewlett-Packard Development Company, L.P. (`policy_builder_guide.pdf`)
- *HP OpenView Select Access 6.2 Network Integration Guide*, © Copyright 2002-2006 Hewlett-Packard Development Company, L.P. (`integration_guide.pdf`)
- *HP OpenView Select Access 6.2 Concepts Guide*, © Copyright 2005 - 2006 Hewlett-Packard Development Company, L.P. (`concepts_guide.pdf`)

Integration Papers for Select Access and vendor-specific technologies are available on the product CDs in the `docs/solutions` folder.

Online help is available with both the Setup Tool and the Policy Builder components.

As part of the Select Access SDK, two other documents are also available with this product:

- *HP OpenView Select Access 6.2 Developer's Tutorial Guide*, © Copyright 2004-2006 Hewlett-Packard Development Company, L.P. (`dev_tut_guide.pdf`)
- *HP OpenView Select Access 6.2 Developer's Reference Guide*, © Copyright 2004-2006 Hewlett-Packard Development Company, L.P. (`dev_ref_guide.pdf`)

For details on how to obtain this SDK, visit HP's Partner Care site (`http://support.openview.hp.com/partner_care.jsp`).

Integration Possibilities

Table 1 lists the various technologies that you can integrate Select Access with. Some technologies integrate seamlessly with Select Access, while others require manual intervention. The technologies that require manual installation and configuration are detailed extensively in this guide.

Table 1 Supported Technologies

Category	Name	Integration Details
Directory servers	Microsoft Active Directory	Transparent
	Microsotf ADAM	Manual
	Sun ONE/iPlanet/Netscape Directory Servers	Transparent
	Critical Path CP Directory	Transparent
	Novell eDirectory	Manual
	Oracle Internet Directory	Manual
	CA eTrust Directory	Manual
	OpenLDAP	Manual
Web servers	Microsoft IIS	Transparent
	Sun ONE/iPlanet/Netscape Web Servers	Transparent
	Apache 2.0	Transparent
	Lotus Domino	Transparent

Table 1 Supported Technologies (cont'd)

Category	Name	Integration Details
Application and portal servers	BEA WebLogic Application Server	Manual
	IBM WebSphere Application Server	Manual
	Tomcat Application Server	Manual
	Sun ONE Application Server/ iPlanet Application Server	Transparent
	Oracle 10g Application Server for Windows only	Transparent
	Plumtree Portal Server	Manual
	Citrix MetaFrame Web Interface	Manual
	Any J2EE compliant server	Manual
Web-enabled applications	Siebel	Manual
	Outlook Web Access	Manual
	Apache Reverse Proxy	Manual
	PeopleSoft	Manual
	MySAP	Manual
Authentication methods	Passwords	Transparent + forms setup
	Windows NT Domains	Transparent + forms setup
	Windows Kerberos	Transparent + forms setup
	Digital Certificates	Transparent
	Registration	Transparent + forms setup
	RSA SecurID	Transparent + forms setup
	Smart cards/2-factor tokens/ RADIUS authentication	Transparent + forms setup
	Custom methods	Manual
Wireless servers	Wireless Application Protocol (WAP) 2.0	Transparent

Chapter Summary

This guide includes the chapters listed in Table 2.

Table 2 Chapter Summary

Chapter	Description
Chapter 2, Directory Server Integrations	Select Access uses a Lightweight Directory Access Protocol (LDAP) v.3-compliant directory server for searching and storing identity information and/or Select Access policy data. To make the adoption of Select Access as easy as possible, HP has included support for several LDAP directory servers. This chapter describes integration with directory servers.
Chapter 3, Transparently Supported Web Server Integrations	Of all the web servers Select Access supports, only three integrate transparently with Select Access: Sun/ Netscape/iPlanet, IIS web servers, Apache 2.0, and Domino web servers. This chapter describes how the transparent suppport of these web servers enables you to protect web-based resources easily.
Chapter 4, Other Apache Server Integrations	Select Access transparently supports Apache 2.0 web servers with its Apache 2 Enforcer plugin. However, Select Access still supports integrations with Apache 1.3.x servers with the Apache 2 Enforcer plugin. This chapter describes these integrations.
Chapter 5, Integrating With .NET Frameworks	To integrate with .NET frameworks, Select Access uses WSE, the .NET class library for building/integrating web services. WSE uses web services protocols, including WS-Security, WS-Routing, DIME, and WS-Attachments. WSE integrates with ASP.NET web services, allowing Select Access to provide a simple way to extend the functionality of these protocols.
Chapter 6, Servlet Engine Integrations	Select Access' servlet Enforcer plugin provides Java Servlet technology a simple, consistent mechanism for extending the security of a servlet engine. Whether the servlet engine is built into an application server, or whether the server requires an add-on module, the servlet Enforcer plugin secures these engines by intercepting identity requests irrespective of how the servlet engine is started.

2 Directory Server Integrations

Select Access uses a Lightweight Directory Access Protocol (LDAP) v.3-compliant directory server for searching and storing identity information and/or Select Access policy data. To make the adoption of Select Access as easy as possible, HP has included support for several LDAP directory servers. This chapter describes integration with directory servers.

By using standards-compliant directory servers and meta directories for access to legacy identity stores, Select Access enables you to synchronize information easily across a globally dispersed networks.

Chapter Overview

Table 1 on page 10 lists the directory servers Select Access currently integrates with. Many of these directory servers are supported seamlessly at runtime. These directories are not discussed at length in this chapter, however, other topics in this chapter may still be of interest to Select Access deployments that transparently integrate with certain directory servers. Topics in this chapter include:

- Integration Overview on page 13
- Directory Topologies: Supporting Multiple Identity Locations on page 14
- Schemas and the Ease-of-Integration on page 19
- Tuning Directory Servers on page 36

Integration Overview

The integration process for directory servers is impacted by different factors. Typical factors that can affect the deployment of Select Access with your directory server include:

- Directory architectures
- Schemas and how transparently they map to Select Access
- Attributes and the indexing of those attributes

Integration Tasks

The following integration tasks are generic to all directory servers, whether or not integration occurs transparently.

Task 1: Set up replication and/or referrals.

If you have a distributed architecture, ensure you have set up replication and referrals correctly. This ensures data remains compatible with Select Access. For details, see Directory Topologies: Supporting Multiple Identity Locations on page 14.

Task 2: Upload schema changes.

If your directory does not transparently integrate with Select Access, ensure you upload schema changes. The process for different directory servers vary. For details, see Schemas and the Ease-of-Integration on page 19.

Task 3: Install and configure Select Access.

Only after you have ensured data remains compatible with Select Access should you begin installing and configuring components on your network. Remember to configure Select Access components to use replication and referrals as described in Directory Topologies: Supporting Multiple Identity Locations on page 14. You know your deployment of Select Access is successful if:

- You can see identities in the Policy Builder.
- You can add access policies, resources, and so on.
- The Policy Validator does not produce any errors when trying to read data to authenticate and authorize identity access for requested resources.

For details, refer to the *HP OpenView Select Access 6.2 Installation Guide*.

Task 4: Index attributes.

As part of your directory server's tuning, indexing is an important feature that you can use to cache frequently used information. This important step can only occur after Select Access has updated your directory server's schema successfully. For details, see Tuning Directory Servers on page 36.

Directory Topologies: Supporting Multiple Identity Locations

Depending on your pre-existing directory system topology, there can be any number of ways you have chosen to structure your identity data. The more profiles you have, the more likely it is that you need to distribute your data across multiple servers. To support distributed directory systems, Select Access offers support for the following topology configurations:

- Replication: the method of making copies of identity or policy data (some or all) to enhance the fault tolerance of your directory system.
- Referrals: the method of redirecting the Select Access component to the directory server that holds the data it requires.

Because Select Access supports multiple identity locations as well as multiple directory servers for replication and referrals, it is important to understand the differences between these support features.

For example, a directory tree might look similar to Figure 1, where mycompany.com is the root of the tree. You can set up the branches of this root such that each branch is a unique identity location with its own set of profiles, groups, and dynamic groups.

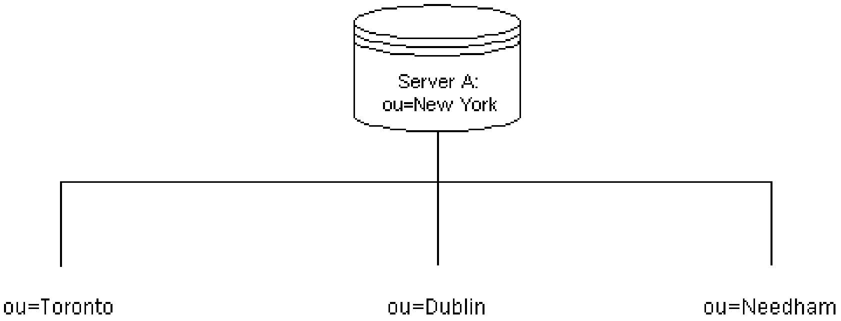


Figure 1 Directory Tree with Multiple Identity Locations

However, rather than storing identity data as branches on this tree, you can also move the data on these branches into separate servers, as shown in Figure 2.

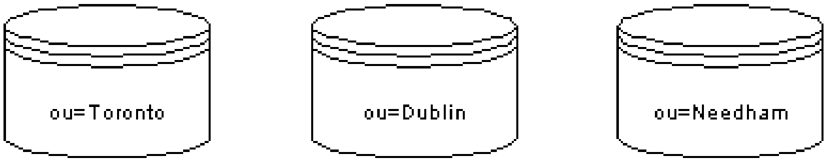


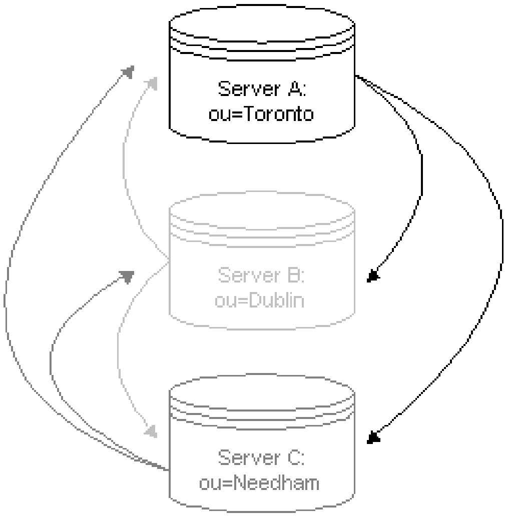
Figure 2 Directory Branches Separated into Different Servers

This configuration still keeps identity locations unique, but gives you the added benefit of being able to separate the identity data onto different servers. In this instance, you could set up your directory servers such that they include a referral directory entry that redirects the Select Access client to the correct location, as shown by Figure 3.

Setting Up Multiple-master Replication with Select Access

Multi-master replication is the only directory server topology that works successfully with Select Access. Single-master replication does not work: if the master goes down, the Policy Validator cannot modify the profile as needed. The Policy Validator must be able to write to the directory server in question.

By defining which directory servers are masters and which are replicas, you delineate when Select Access components can write to or read from any given instance of a directory server. Select Access supports multiple-master replication because it minimizes downtime if a single server fails.



Directory servers
smart referral entries
to other servers:

Server A points to
ou=Dublin and
ou=Needham

Server B points to
ou=Toronto and
ou=Needham

Server C points to
ou=Dublin and
ou=Toronto

Figure 3 Referrals: Multiple Directory Servers with Multiple Identity Locations

To design your topology for replication, you might take these servers and create replicas of them on different hosts, as shown in Figure 4. These replicas must be readable and writable.

master:
ou=Toronto
replica:
ou=Toronto

Figure 4 Replication: Duplicated Multiple Directory Servers

To set up replicated directory servers for use with Select Access

1. Set up administration credentials so that they share the same usernames and passwords. Ensure these credentials match the one you configured in Select Access.
2. Set low values for connection timeouts on the host machine's operating system for the:
   - Administration server
   - Policy Validator

   That way, if one of the directory server's host machine fails, the Policy Validator does not spend a lot of time trying to connect to a machine that is unreachable.

   > If you have not set an aggressive connection timeout on the operating system, set the Enforcer plugin's **Wait for Validator reply** parameter on the **Tuning** setup screen to a larger value to compensate. For details, see Chapter 7, Configuring the Policy Validator, in the *HP OpenView Select Access 6.2 Installation Guide*.

3. Ensure the schemas on replicated servers are the same. Without consistency, replication operations fail. Do this by:
   - Checking attributes across all servers to ensure they match
   - Verifying version numbers of all directory server software to ensure they are the same
4. Make all replicated directories writable. This is particularly important if you allow your identities to self-manage their profiles.

5 Maintain schema consistency efforts you initiated in step 3, by applying Select Access schema changes consistently. HP recommends the following procedure to maintain consistency across your system:

- Upload Select Access schema changes and update identity information on the master exclusively.
- When complete, replicate the changes across remaining servers immediately.

> ➤ For Policy Validators that require up-to-date information, the speed with which this step is implemented is critical to protect your corporate resources. For example, inconsistent data can cause the Policy Validator to deny access to a identity who would normally be allowed to access a given resource.

To use replicated servers with Select Access

1 In the Administration server, populate the replication parameters by choosing **Custom** and configuring the **Replicated Directory Servers** setup screen. For details, see Chapter 5, Configuring the Administration Server, in the *HP OpenView Select Access 6.2 Installation Guide*.

2 In the Policy Builder, set up identity location to prevent identity authentication ambiguities from occurring when the Policy Validator caches profiles.

Select Access does not allow you to create identity locations that have overlapping root DNs. Overlapping root DNs occur when you try using:

- A child of an existing identity location
- The parent of an existing identity location
- A root DN of a different directory with the same name

For example, if you add a identity location with a DN of ou=mycompany.com, ou=users, but you have subunits for each of your regional offices, you cannot add subsequent identity locations like:

```
o=mycompany.com, ou=users, ou=americas

o=mycompany.com, ou=users, ou=europe

o=mycompany.com, ou=users, ou=asia
```

Instead, you would have to either remove the ou=mycompany.com, ou=users location, or rearchitect your identity locations in one of the following ways:

- By adding new directory servers. For example:

```
o=americas.mycompany.com, ou=users

o=europe.mycompany.com, ou=users

o=asia.mycompany.com, ou=users
```

- By creating new top-level identity sources. For example:

```
o=mycompany.com, ou=americas_users

o=mycompany.com, ou=europe_users

o=mycompany.com, ou=asia_users
```

Setting Up Smart Referrals

Smart referrals differ from default referrals in that they are more dynamic and use actual entries in the directory to point to the directory server. Select Access does not support default referrals. If you are currently using default referrals, you need to set up your directory servers to use smart referrals instead.

A typical referral, including default referrals, use the LDAP URL syntax:

```
ldap://<hostname>:389/<DN>
```

where:

- `<hostname>` is the name of the machine hosting the directory server
- `<DN>` is either:
  - — The base DN when performing lookups
  - — The target DN when performing events such as additions, deletions, or modifications

Using the example in Figure 3, a search referral to Server B might look like this:

```
ldap://ServerB.mycompany.com:389/ou=Dublin
```

➤ Select Access does not implement the `ldap_rebind` code. This means that the referred LDAP server must have the same credentials as the LDAP server that contains the referral.

To use smart referrals with Select Access

1 Create a folder that uses a subdirectory of the Identities Tree. For example, if you created a folder called “Contract Employees”, the folder would seamlessly list the referred profiles without administrators not noticing that a referral is happening.

2 Create profiles that act as a smart referral. Define these entries with:

- The `ref` attribute. This attribute takes the appropriate LDAP URL value to identify the true host of the data requested.
- The `referral` objectclass.

➤ If you are using an iPlanet directories, ensure that the `ou` attribute contains the same value on both the referred and the referring servers.

3 In the Policy Validator, add the optional bootstrap parameter called `ldapUseReferral`, which you must manually include in the configuration file. This parameter specifies whether or not the directory server generates referrals. The value uses the LDAP URL to help the Policy Validator to find the appropriate directory on the network.

4 To avoid having timeouts exceeding a Policy Validator wait for reply time, ensure that the operating system of the directory host uses aggressive timeout values. If you cannot change system timeout values, HP suggests that you configure the **Wait for Validator reply no more than X seconds** (a **Tuning** setup screen parameter) to a high value.

Schemas and the Ease-of-Integration

Transparent integrations occur because certain directory schemas map more seamlessly with Select Access than others. In instances where Select Access requires updates to the schema, you must upload changes to the directory manually to obtain desired functionality in Select Access components. This chapter primarily documents how to integrate Select Access on these directories.

Schemas Requiring Manual Changes and Specific Integration Concerns

If your directory server requires schema changes or have known issues that require you reconfigure your server, you must do this *before* you install and run Select Access.

Schemas may need periodic updates as you upgrade to newer versions. The procedure for uploading schema changes for the first time can differ from updating schemas for upgraded functionality. Each directory documented in this chapter document differences accordingly, if applicable.

Only directories that require manual schema changes or have known issues that require your attention are listed in Table 3. For a complete list of supported directory servers, see Supported LDAP Directory servers in *HP OpenView Select Access 6.2 Installation Guide*.

Table 3 Directory-Specific Details

Directory name and version	User schema changes?	Policy schema changes?	For first-time installs	For upgrades	For known issues
The Microsoft Active Directory Servers for Windows 2000 with SP4 and for Windows 2003	transparent	transparent	N/A	N/A	
The NDS eDirectory 8.7.3 Server	transparent	transparent			
The Critical Path 4.2 Directory Server			page 21		
The Sun ONE 5.2 and Netscape 6.01 Directory Servers	yes	no	page 22	N/A	
The Microsoft Active Directory Application Mode (ADAM) Server for Windows 2003			page 22	N/A	

Table 3 Directory-Specific Details (cont'd)

Directory name and version	User schema changes?	Policy schema changes?	For first-time installs	For upgrades	For known issues
The Oracle Internet Directory 9.2 Server	yes	yes	page 27	N/A	page 29
The CA eTrust 8 Directory Server on page 29	yes	yes	page 29	page 32	page 33
The OpenLDAP 2.2.23 Directory Server	yes	yes	page 33	N/A	

The Microsoft Active Directory Servers for Windows 2000 with SP4 and for Windows 2003

Active Directory Servers (ADS) are transparently supported by Select Access. However, certain issues with ADS require that you make certain configuration choices:

- You may need to make the ADS schema writable. For details, see To make the ADS schema writable on page 21.
- If you make `givenname` a preferred and active attribute in the Select Access system, note that it has a maximum unicode character limit of 64. For details on enabling attributes in Select Access, see To select the preferred identity attributes in the *HP OpenView Select Access 6.2 Policy Builder Guide*.
- If you want to use Select Access' password management feature, enable SSL on ADS. Otherwise you cannot create user profiles with passwords. Additionally, make sure that passwords meet the criteria set out by ADS; otherwise, the password cannot be created when you enter details in the **New Identity** dialog box.
- To allow Select Access components full schema access privileges, ensure Select Access logs on as local user by configuring the correct information in the Administration server.
- Microsoft has placed a size limit on LDAP records. This prevents you from activating all available attributes. However, this is typically not a problem since most deployments do not require all user attributes activated.
- The user object class is `User` not `inetOrgPerson`, which is the object class used by all other directory servers. The difference in user object class impacts the following components and features, because the number and the types of user attributes between these two classes vary:
  - — How the Policy Validator performs password-based authentication
  - — How the Policy Validator registers new users
  - — Password management of users

➤ If an attribute is not available in the `User` class, you can add it by clicking **Advanced** of the **User properties** dialog box, and modifying the **Attributes** tab accordingly. ADS does not allow you to add an attribute that is contrary to the schema definition for user entries.

- ADS does not allow you to create an `ADS group` within another ADS group. This is because this group type is a container for users only. It will not allow you to add other groups of the same native type.

- An ADS issue causes directory referrals on an SSL system to refer to a non-SSL location, which causes referrals to fail on Select Access. This issue is expected to be fixed in a future release of ADS.

To make the ADS schema writable

1. Run the `regedit` application.
2. Open the following folder:

   ```
   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Parameters
   ```

3. In the right pane, right-click in an empty area of the window. Select **New** → **DWORD Value.** An empty DWORD entry appears in the **Name** column.
4. Type `Schema Update Allowed` and click **Enter**.
5. Double-click the value name you typed in Step 4. The **Edit DWORD value** dialog appears.
6. In the **Value data field**, type `1` and click **OK**. This updates the `<SA_install_path>\schema\Active-Directory\adupdate.reg` file and creates an entry similar to the one shown below:

   ```
   [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
   NTDS\Parameters] "Schema Update Allowed"=dword:00000001
   ```

   This file is loaded into the registry on the machine where ADS is running, which makes it possible to update the ADS schema from the Setup Tool.

The NDS eDirectory 8.7.3 Server

eDirectory servers are transparently supported by Select Access, however, certain issues exist with eDirectory. LDAP messages larger than 64K are truncated in eDirectory, which causes problems with other Select Access components like the IIS Enforcer plugin. In addition, Novell eDirectory rejects any DN that is longer than 256 characters with LDAPException. These errors are not caused by Select Access but by the character limitations.

The Critical Path 4.2 Directory Server

Whether you are installing Select Access against a CP server for the first time, or whether you are upgrading a previous installation of Select Access, take note of the following and ensure your configuration of either CP and/or Select Access meets the following criteria:

- Operator handling produces different results on CP than other directories, therefore the LDAP attribute decision point may not return desired results. In particular, the less than (<), greater than (>), less than or equal to (<=), and greater than or equal to (>=) operators tend to be the most inconsistent.
- Reconfigure index node values in order to avoid generating an error in the Policy Builder. The default index node value is too low. To set the correct maximum CP index node value:
  1. Open the following file in a text editor of your choice:

     ```
     <CP_install_path>/ds.properties
     ```

  2. Locate or add the following parameter and ensure that it has the corresponding value:

     ```
     directory.indexNodeMax=524288
     ```

  3. Restart your directory server to ensure it uses the new parameter value.

- In CP's configuration file, change the `Admin Size Limit` parameter to something much higher than 50. This limit controls the amount of data returned by CP in response to a search request. If the limit is set too low, the following error occurs:

  ```
  Administrative limit exceeded
  ```

  ➤ HP recommends a value of at least 1000. If the size limit is too small, you can experience unpredictable errors with data signing and other important features that require a high administration size limit.

The Sun ONE 5.2 and Netscape 6.01 Directory Servers

If you are using these servers, you need to enable and disable specific plugins to use these directories. Once you have correctly setup the right combination of plugins, no additional configuration is required.

To enable and disable specific plugins

1. Launch and log into the Sun ONE or Netscape Console.
2. Click the **Servers and Applications** tab and expand the **Server Group**.
3. Select **Directory Server** and click **Open**. The **Directory Server** window appears.
4. Click the **Configuration** tab and expand the **Plugins** tree entry.
5. Do the following:
   a. Click the **uid uniqueness** entry. (Required.) In the Configuration screen, select **Enable plug-in** and then click **Save**.
   b. Click the **7-bit check** entry. (Optional in deployments that save something other than non-ascii values.) In the Configuration screen, select **Disable plug-in** and then click **Save**.

The Microsoft Active Directory Application Mode (ADAM) Server for Windows 2003

To ensure Select Access and ADAM function properly as a unit, you must follow a specific series of steps to integrate them correctly. Table 4 summarizes the steps you need to perform to configure and delegate all authentication and authorization responsibilities to Select Access.

Table 4 Integrating Select Access with ADAM for Identity and Policy Data

Setup Task	Details
Step 1: Install ADAM and create a new partition instance.	1 Run `adamsetup.exe` from a Command Prompt or double-click `adamsetup.exe` in Windows Explorer. 2 Create a new partition instance. 3 Verify that you have a new partition instance. Select Access needs this partition instance to load its schema and policy directory entries.

Step 2: Add the Active Directory and Select Access schemas.
Note: If you already added the Active Directory schema to your ADAM directory server, you may only need to add the Select Access schema. This depends on how much of the entire set of the Active Directory schema you added and how much customization you did to it. You can examine the `ad_schema.ldf` file and compare it to your current ADAM schema.
The path to the schema files is: `\Select Access\Schema\ADAM`
	1 Back up your directory server. Do this before you add any new schema elements.
	2 To go to a Command Prompt, do one of the following:
	• Click **Start** → **Programs** → **ADAM** → **ADAM Tools Command Prompt**.
	• At a DOS Command Prompt, change directories to `\Windows\ADAM`.
	3 At the Command Prompt, type `ldifde.exe` to set the Active Directory and Select Access schemas in this order:
	• `ad_schema.ldf` (Active Directory schema)
	• `sa_schema.ldf` (Select Access schema)
	ad_schema.ldf Example
	`ldifde -i -f c:\temp\adam\ad_schema.ldf -c "cn=schema,cn=configuration,dc=x" "#schemaNamingContext" -s glacier.can.hp.com -t 50006`
	sa_schema.ldf Example
	`ldifde -i -f c:\temp\adam\sa_schema.ldf -c "cn=schema,cn=configuration,dc=x" "#schemaNamingContext" -s glacier.can.hp.com -t 50006`

Step 3: Set up ADAM to use SSL so you can use Self-Registration and/or manage a user's password through the Select Access Policy Builder.	For details, refer to your Microsoft documentation. Although it is not recommended, you can change the directory server behavior using the `dsmgmt.exe` command in the ADAM Tools Command Prompt. Refer to the ADAM technical reference manual for instructions.

Table 4 Integrating Select Access with ADAM for Identity and Policy Data

Setup Task	Details
Step 4: If no users/groups exist, create an entry that uses the `organizationUnit` object. Note: By default, `organizationUnit` is allowed under `country` (`c`), `organization` (`o`), `organizatinalUnit` (`ou`), and `domainDNS` (`dc`). For information about adding superiors, refer to the ADAM information page at: http://www.microsoft.com/windowsserver2003/adam/ADAMfaq.mspx. Note: If users/groups exist, skip to the next step.	1 Click **Start** → **Programs** → **ADAM** → **ADAM ADSI Edit**. 2 Using the `organizationUnit` object, create a group named `Users`.
Step 5: Create an administrator profile for Select Access. This profile allows the Select Access system to log into ADAM as needed.	Creating an Administrator Profile for Select Access on page 25.
Step 6: If you want to use the ADAM directory as both your Select Access Policy Store and as a source of user data, configure the Administration server and directory server.	1 Run the Setup Tool. 2 Click **Next** on the Welcome screen. 3 Click **Next** on the list of detected components screen. The **Administration Server** setup wizard appears. 4 Click **Configure**. The **Administrator** setup screen appears. 5 Configure the Select Access administrator credentials and click **Next**. The **Directory Server** setup screen appears. 6 Use the credentials of your newly-created “manager” profile. Note: In the **Login Name** field, type the user ID you configured in Step 5. For example, “`cn=manager`”. For details, see Step 4 in To create an Administrator Profile for Select Access on page 25. 7 In the **Password** field, type the password you configured for that identity. Select Access uses these credentials each time it logs into the ADAM directory.

Table 4 Integrating Select Access with ADAM for Identity and Policy Data

Setup Task	Details
Step 7: If you only want to use ADAM as a source of user data, configure the new user store using the Policy Builder.	1 Start the Policy Builder. 2 Click **Tools** → **Identity Location Configuration**. 3 Click **Add**. 4 Type the detailed information for the ADAM directory and click **OK**. For more information, see the *HP OpenView Select Access 6.2 Policy Builder Guide*.

Creating an Administrator Profile for Select Access

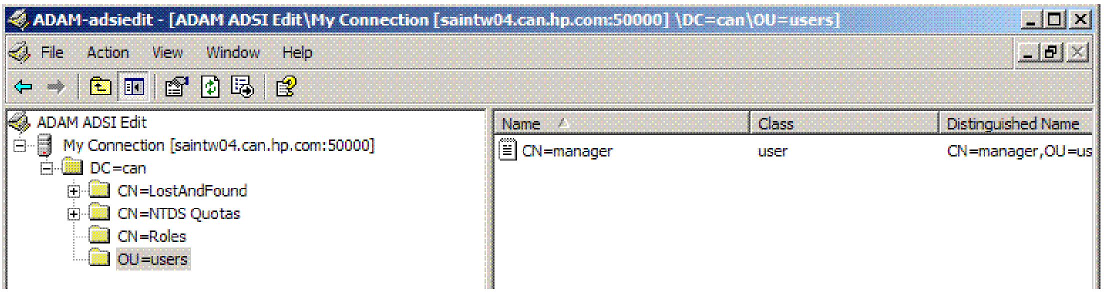
An entity profile for Select Access is not part of ADAM by default. The most important elements are:

- The role to which it is assigned (role)
- The login credentials Select Access uses to log in

An administrator profile gives Select Access the ability to act as the ADAM administrator. This allows Select Access to log in and manage user profiles and authentication/authorization policies accordingly. The creation of this profile is a vital step in the integration of these two products.

To create an Administrator Profile for Select Access

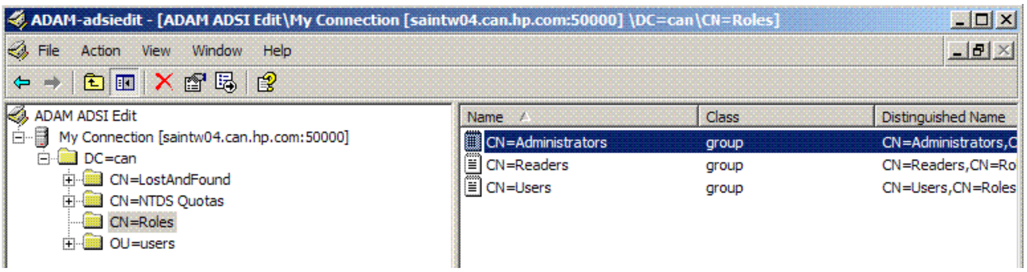
1 Create a `cn=manager` entry in the `OU=users` container.


ADAM-adsiedit - [ADAM ADSI Edit\My Connection [saintw04.can.hp.com:50000] \DC=can\OU=users]
File Action View Window Help
ADAM ADSI Edit
My Connection [saintw04.can.hp.com:50000]
DC=can
CN=LostAndFound
CN=NTDS Quotas
CN=Roles
OU=users

Name	Class	Distinguished Name
CN=manager	user	CN=manager,OU=us

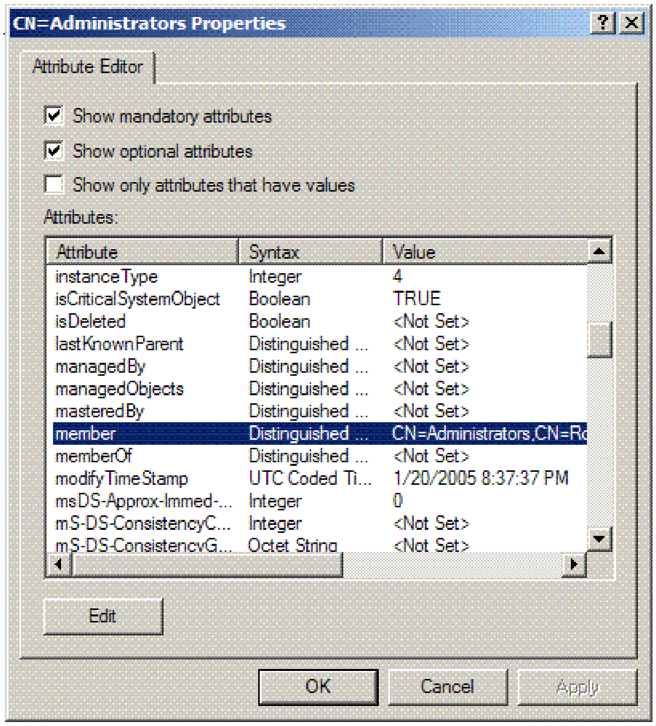
2 Do the following to add this entity profile as a member of the `Administrators` role:

a Click **CN=Roles**.


ADAM-adsiedit - [ADAM ADSI Edit\My Connection [saintw04.can.hp.com:50000] \DC=can\CN=Roles]
File Action View Window Help
ADAM ADSI Edit
My Connection [saintw04.can.hp.com:50000]
DC=can
CN=LostAndFound
CN=NTDS Quotas
CN=Roles
OU=users

Name	Class	Distinguished Name
CN=Administrators	group	CN=Administrators,C
CN=Readers	group	CN=Readers,CN=Ro
CN=Users	group	CN=Users,CN=Roles

b Right-click the **cn=administrator** entry, and then click **Properties**. The **CN-Administrator Properties** dialog box appears.


CN=Administrators Properties

Attribute Editor

☑ Show mandatory attributes

☑ Show optional attributes

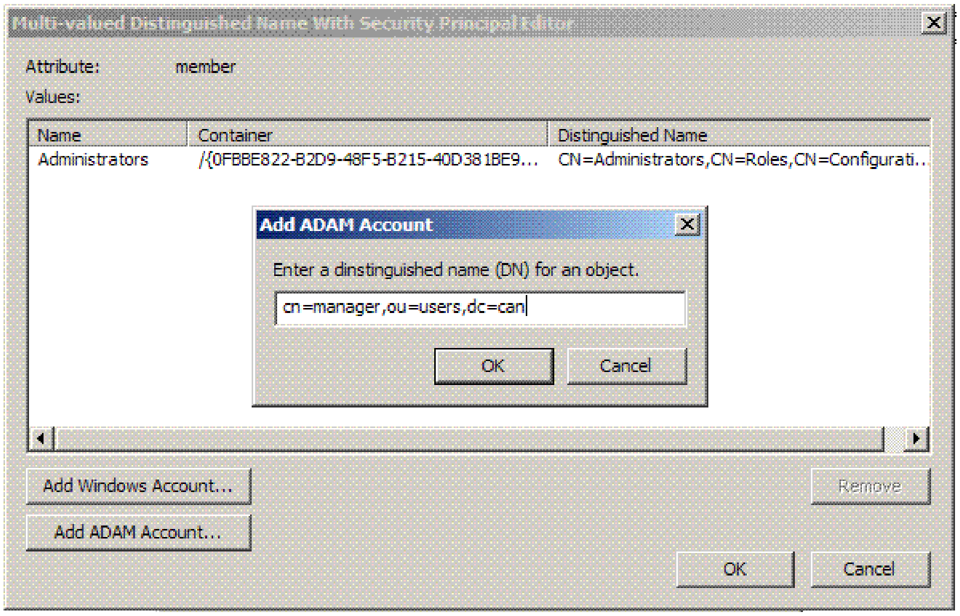
□ Show only attributes that have values

Attributes:

Attribute	Syntax	Value
instanceType	Integer	4
isCriticalSystemObject	Boolean	TRUE
isDeleted	Boolean	<Not Set>
lastKnownParent	Distinguished ...	<Not Set>
managedBy	Distinguished ...	<Not Set>
managedObjects	Distinguished ...	<Not Set>
masteredBy	Distinguished ...	<Not Set>
member	Distinguished ...	CN=Administrators,CN=Rc
memberOf	Distinguished ...	<Not Set>
modifyTimeStamp	UTC Coded Ti...	1/20/2005 8:37:37 PM
msDS-Approx-Immed-...	Integer	0
mS-DS-ConsistencyC...	Integer	<Not Set>
mS-DS-ConsistencyG...	Octet String	<Not Set>

c In the **CN-Administrator Properties** dialog box, click **Edit**. The **Multi-valued Distinguished Name With Security Principal Editor** dialog box appears.

d In the **Multi-valued Distinguished Name with Security Principal Editor** dialog box, click **Add ADAM Account**. To set your newly created `cn=manager` entry as an administrator role, edit the members attributes and type the required information.


Multi-valued Distinguished Name With Security Principal Editor

Attribute: member

Values:

Name	Container	Distinguished Name
Administrators	/{0FBBE822-B2D9-48F5-B215-40D381BE9...	CN=Administrators,CN=Roles,CN=Configurati...

Add ADAM Account

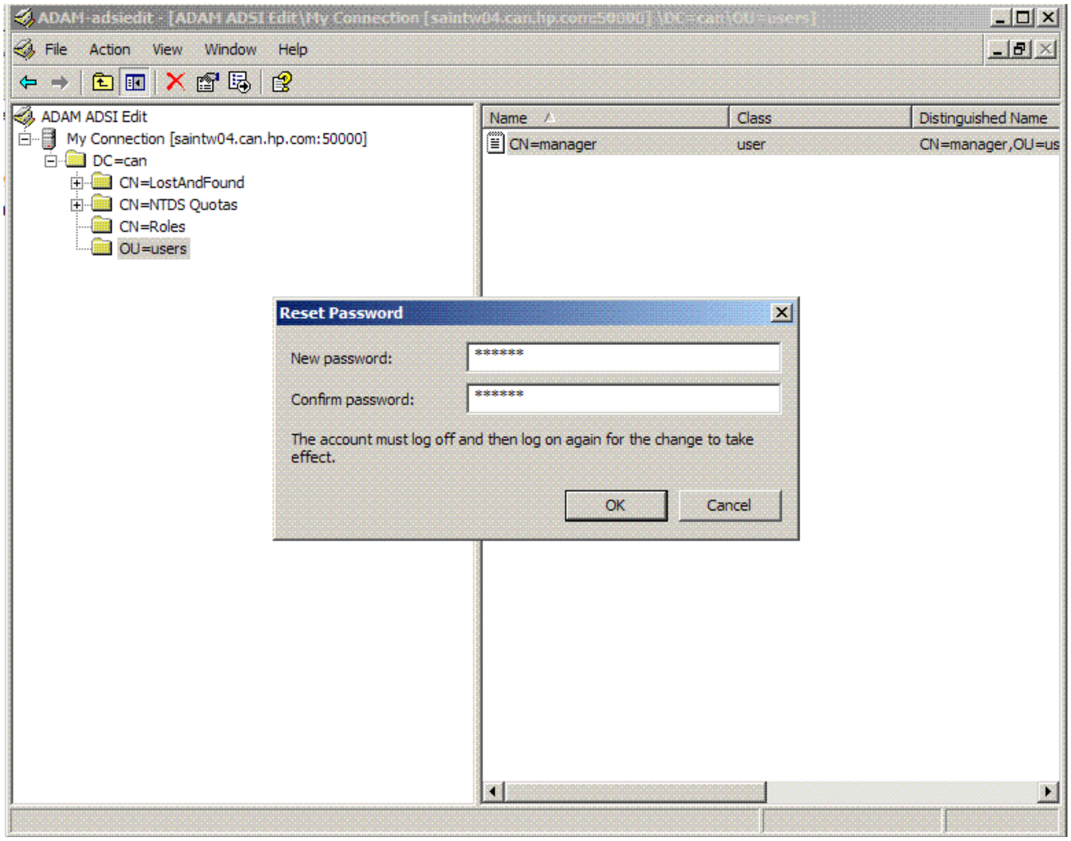
Enter a dinstinguished name (DN) for an object.

cn=manager,ou=users,dc=can

3 Do the following to create a password for the entity profile for Select Access.

a Right-click the newly created `cn=manager` entry in the `OU=users` directory entry.

b Click **Reset Password**. The **Reset Password** dialog box appears.


ADAM-adsiedit - [ADAM ADSI Edit\My Connection [saintw04.can.hp.com:50000] \DC=can\OU=users]
File
Action
View
Window
Help
ADAM ADSI Edit
My Connection [saintw04.can.hp.com:50000]
DC=can
CN=LostAndFound
CN=NTDS Quotas
CN=Roles
OU=users
Reset Password
OK
Cancel

Name	Class	Distinguished Name
CN=manager	user	CN=manager,OU=us

New password: ******

Confirm password: ******

The account must log off and then log on again for the change to take effect.

c Type and confirm a new password.

The Oracle Internet Directory 9.2 Server

This directory server requires manual intervention before the schema uploads correctly. Table 5 outlines the steps you must consider.

Table 5 Uploading Schema Changes

Upload task	Required for	For details, see
1 Configure SSL on the Oracle Internet Directory (OID) server.	User schemas and Policy schemas	To configure SSL settings on page 28
2 Modify the naming context attributes to add those required by Select Access.	User schemas and Policy schemas	To modify the naming context attribute on page 28
3 Create object classes and configure properties as required by Select Access.	User schemas and Policy schemas	To create required object classes properties on page 29

Table 5 Uploading Schema Changes (cont'd)

Upload task	Required for	For details, see
4 Rerun the Setup Tool and reconfigure the policy and/or identity data locations on this directory server.	User schemas and Policy schemas	*HP OpenView Select Access 6.2 Installation Guide*
5 Restart any directory server instances so changes are uploaded.	User schemas and Policy schemas	OID documentation
6 Use Oracle Directory manager to change the default encryption algorithm used from MD4 to: — MD5 — SHA-1	Policy schemas	OID documentation

To configure SSL settings

1 Log onto Oracle Internet Directory. The **Oracle Directory Manager** appears.

2 In the **System Objects** pane, expand **Server Management**, then **Directory Server**, and select **Default Configuration**.

3 Click the **SSL Settings** tab.

4 Select the **SSL Enable** check box at the top of the tab.

5 In the **SSL Authentication** field, select **SSL Server Authentication** from the drop-down list.

6 Click **Apply**.

To modify the naming context attribute

1 From the **Oracle Directory Manager**, in the **System Objects** pane, expand `Oracle Internet Directory Servers` and select the directory server on which you want to specify a naming context. The corresponding tabs for that directory server appear in the right-hand pane.

2 Click the **System Operational Attributes** tab.

3 If the **Naming Contexts** field is blank, fill it in one of the following two ways:

- Type the topmost DN of the naming context you want to publish. For example, `o=mycompany.com`.
- Click **Browse** to open a search window and browse for the naming context you want to publish.

  Select Access uses the **namingContexts** attribute to locate the root entry of the directory server. By default, this attribute is blank.

4 In the **Password Encryption** field, select the type of password hashing you want to use. Your options are: `MD4`, `MD5`, `No encryption` and `SHA`, `UNIX`. Select Access supports all algorithm options in the drop-down list except for MD4.

> Select Access password reset and password management doesn't support the UNIX CRYPT algorithm.

5 Click **Apply**.

To create required object classes properties

1. From the **Oracle Directory Manager**, in the **System Objects** pane, expand **Oracle Internet Directory Servers**.
2. Right-click the **Entry Management** tree entry and then click **Create**.
3. In the **Distinguished Name** field, enter the topmost DN that you used for the naming context (for example, `o=mycompany.com`). For details, see To modify the naming context attribute on page 28.
4. In the **Object classes** section, click **Add**. A list of object classes appears.
5. Click the `organization` object class and then click **Select**.
6. Click **Add**, click the `top` object class and then click **Select**.
7. In the **Mandatory Properties** section, type in the DN (for example, `mycompany.com`).
8. Click **OK**.

Known Issue with OID

Whether you are installing Select Access against an OID server for the first time, or whether you are upgrading a previous installation of Select Access, take note of the following and ensure your configuration of either OID and/or Select Access meets the following criteria:

- When using an OID in a multi-Policy Validator deployment, an error is generated when a Policy Validator tries to update a nonce secret. All nonces are deleted and authentication fails via nonces, requiring the end user to login again. Because attribute matching fails with binary attribute values on OID, the Policy Validator does not delete old nonce secrets on this server, but instead just adds new ones.

  ➤ This is an OID issue and not an Select Access issue. OID is not a recommended platform in this environment.

- Versions of supported JDKs vary between Select Access and OID. Select Access supports JDK 1.4.1. OID, however, only supports JDK 1.3.1. Because international special characters are not supported in this lower version of the JDK, the oidadmin GUI cannot display the Japanese special characters. This limitation, causes the oidadmin GUI to display these characters as '?'.

The CA eTrust 8 Directory Server

Integrating Select Access with eTrust Directory primarily involves modifying the Directory System Agent (DSA) group file to include the Select Access-specific schema. A group file references one to many configuration files. These configuration files contain the schema that is loaded into the directory server.

All the configuration files specified in a DSA group file are loaded into the DSA when the server is started. You need to add the Select Access schema configuration file to the DSA group file so that the Select Access schema is available to it.

If this is your first time installing Select Access, Table 6 outlines the steps you must consider.

Table 6 Uploading Schema Changes

Upload task	Required for	For details, see
1 Copy the eTrust schema script to an eTrust location.	User schemas and Policy schemas	To copy the eTrust schema script on page 30
2 Modify the directory server schema to make `uniqueMember` an optional attribute. This modification makes allows you to create groups without assigning membership to that group (which is sometimes required when authenticating registering new, and therefore currently unknown, users).	User schemas only	To make uniqueMember an optional attribute on page 30
3 Modify the group file for the DSA that you want Select Access to reference.	User schemas only	To change the group schema file on page 31
4 Because eTrust's default value for this parameter is too low for the needs of Select Access, increase the `max-op-size` parameter from its default value.	User schemas and Policy schemas	To modify the maximum operation parameter on page 32
5 Start the DSA to load the modified group file.	User schemas only	To load the modified DSA group file on page 32

To copy the eTrust schema script

1. Locate the `SAeTrustSchema.dxc` file. By default, all schema-related files are installed to `<install_path>/schema/CA-eTrust/`.
2. Copy the script to the following eTrust location:

```
<eTrust_install_path>/Directory/dxserver/config/schema
```

To make uniqueMember an optional attribute

1. Locate the `x500.dxc` file in the following eTrust location:

```
<eTrust_install_path>/Directory/dxserver/config/schema
```

2. Back up the `x500.dxc` file.
3. Open the `x500.dxc` file in a text editor.
4. In the `schema set object-class standardObjectClass:17 = {` section of the file, delete `uniqueMember` from the `must-contain` section and then add it to the `may-contain` section, as shown in the example below.

➤ If you have CA ETrust 8.0 installed, `uniqueMember` is already in the `may-contain` section.

```
schema set object-class standardObjectClass:17 = {
        name = groupOfUniqueNames
        subclass-of top
        must-contain
            commonName
        may-contain
            uniqueMember,
            description,
        organizationName,
            organizationalUnitName,
            owner,
            seeAlso,
            businessCategory
};
```

5 Save the modified `x500.dxc` file.

To change the group schema file

1 Open your group schema file from the following eTrust location:

```
<eTrust_install_path>/dxserver/config/schema/<group_file.dxg>
```

where `<group_file.dxg>` is the name of your group file.

> You can determine the name of the schema group file by looking in the schema section of the initialization file, which begins with the `# schema` line. Initialization files are located in the following location:
>
> `<eTrust_install_path>/dxserver/config/servers`.
>
> For example, in the sample initialization file shipped by eTrust (called `democorp.dxi`), the group file is represented as follows:
>
> ```
> source “../schema/default.dxg”;
> ```

2 Add the following line to the end of the group schema file, as shown the code sample below.

```
source “SAeTrustSchema.dxc”;
```

This adds the Select Access-specific eTrust schema script to the group file. For example:

```
source “x500.dxc”;
source “cosine.dxc”;
source “mhs.dxc”;
source “quipu.dxci”;
source “umich.dxc”;
source “inetop.dxc”;
source “dxserver.dxc”;
source “jndi.dxc”;
source “unspsc.dxc”;
source “SAeTrustSchema.dxc”;
```

3 Save the modified `.dxg` file.

To modify the maximum operation parameter

1 Open the service limits configuration file from the following eTrust location:

```
<eTrust_install_path>/config/limits/default.dxc
```

2 Find the following entry:

```
set max-op-size = 200
```

3 Change this entry as follows:

```
set max-op-size = 2000
```

> HP recommends a value of at least `2000`. This increase allows Select Access to function seamlessly. If the size limit is too small, you can experience unpredictable errors with data signing and other important features that require a high operation size limit.

4 Save the modified `.dxc` file.

To load the modified DSA group file

1 If your DSA is running, stop the service from the Windows services applet. The DSA service name has the following syntax:

```
eTrust Directory - <DSA name>
```

For example, the democorp sample DSA service name is `eTrust Directory - democorp`.

2 Restart your DSA. The schema upload was successful if the DSA starts without any errors.

3 Log into eTrust Directory via the Policy Builder and ensure that your Identities Tree gets populated.

Upgrading eTrust with Version 6.2 Schema Changes

If you are running a version of Select Access previous to 6.2, you need to upgrade Select Access' schema changes to `ldapmodify` takes the name of the Select Access LDIF script as a parameter functionality this version of Select Access requires. This ensures that policy data can be used and/or manipulated by all Select Access components, without causing adverse side effects to the directory server, the data, or the security of your network. The following topic lists steps you must consider.

To upgrade Select Access schema files for eTrust

1 Locate the `SAeTrustSchema.dxc` file. By default, all schema-related files are installed to *`<install_path>`*`/schema/CA-eTrust/`.

2 Copy the script to the following eTrust location:

```
<eTrust_install_path>/Directory/dxserver/config/schema
```

3 Load the modified DSA group file:

a If your DSA is running, stop the service from the Windows services applet. The DSA service name has the following syntax:

```
eTrust Directory - <DSA name>
```

For example, the democorp sample DSA service name appear as follows:

```
eTrust Directory - democorp
```

b Restart your DSA. The schema upload was successful if the DSA starts without any errors.

c Log into eTrust Directory via the Policy Builder and ensure that your Identities Tree gets populated.

Known Issue for eTrust

Whether you are installing Select Access against an eTrust server for the first time, or whether you are upgrading a previous installation of Select Access, take note of the following and ensure your configuration of Select Access meets the following criteria:

- Umlauts in the name of authentication services are not supported due to case-sensitive string types in eTrust.

The OpenLDAP 2.2.23 Directory Server

The OpenLDAP directory server requires manual intervention before the schema uploads correctly. If this is your first time using Select Access with this directory, Table 7 outlines the steps you must perform.

Table 7 Uploading Schema Changes

Upload task	For details, see
1 Stop the OpenLDAP directory and back up all the contents of your directory before continuing.	OpenLDAP documentation
2 Modify properties in OpenLDAP's configuration file to allow Select Access to read and write to the directory.	To modify the OpenLDAP schema and access on page 34
3 To activate a password to restrict directory access, you need to create a root DN entry manually. This is because OpenLDAP does not support the dynamic creation of root DNs. Note that you are not adding the entry to the directory with this step. You are only creating the definition for it.	To create a root DN entry on page 35
4 Start the OpenLDAP directory.	OpenLDAP documentation

Table 7 Uploading Schema Changes (cont'd)

Upload task	For details, see
5 Modify the directory server schema to make uniqueMember an optional attribute. This modification allows you to create groups without assigning membership to that group (which is sometimes required when authenticating registering new, and therefore currently unknown, users).	To make uniqueMember an optional attribute on page 36
6 Check that the server is running and that you have configured it correctly, by running `ldapsearch`.	OpenLDAP documentation
7 Configure the Select Access Administration server so that it uses the root DN you created.	*HP OpenView Select Access 6.2 Installation Guide*

To modify the OpenLDAP schema and access

1. Open `sldap.conf`. Depending on your operating system, the default path is:
   - For UNIX: `/usr/local/etc/openldap`
   - For Windows: `/openldap`
2. Ensure you are already using the following four OpenLDAP schemas.
   - `core.schema`: an OpenLDAP-specific file containing "core" schema definitions.
   - `cosine.schema`: contains LDAPv3 schema derived from the X.500 COSINE "pilot" schema.
   - `inetorgperson.schema`: holds attributes about people. The attributes it holds were chosen to accommodate information requirements found in typical Internet and Intranet directory service deployments.
   - `nis.schema`: contains definitions for RFC2307, using LDAP as a Network Information Service.

   ➤ Some standard objects that are defined by OpenLDAP in the `nis.schema` file have been removed from the Select Access `sa.schema` file. You must copy the `sa.schema` file from the Select Access `schema/openldap` directory to the OpenLDAP schema directory and modify the `slapd.conf` file to enable the `nis.schema` file, if it is not already included.

   These files are included with OpenLDAP and make its schema conform to RFC1274 and RFC2798 standards. Without these files, you will not be able to integrate Select Access with OpenLDAP directory.
3. Copy the schema from the product CDs in the `<install_path>/schema/OpenLDAP` folder and save it to the following directory:

   ```
   <OpenLDAP_install>\schema
   ```
4. Append Select Access' schema to the end of this list of files used by OpenLDAP.

5 Configure the suffix, root DN, and password as properties of this file, with the following entries:

```
suffix   “dc=<mycompany>, dc=com”
rootdn   “cn=Manager, dc=<mycompany>, dc=com”
rootpw   <manager_password>
```

Remember the rootDN and password you configure. You will need this information to configure the Select Access Administration server.

6 Save the changes you made to the `slapd.conf` file.

7 To implement these changes, restart your directory server.

To create a root DN entry

1 Do one of the following:

- If you are installing and running OpenLDAP for the first time: With a text editor of your choice, create a new file called `entries.ldif`. Save this file to suitable location for your deployment environment.
- If you already have deployed OpenLDAP in your organization: Open the `entries.ldif` file in a text editor of your choice, from location you last saved it to.

2 Ensure the file includes an LDIF-compatible entry for your root domain name. For example:

```
dn: dc=<mycompany>, dc=<com>
dc: <mycompany>
objectClass: top
objectClass: domain
```

Replace `<mycompany>` and `<com>` with the appropriate domain components of your domain name. Certain domains contain more components than others. For example, `gov`, `edu`, `ca`, and so on, are all components you must add.

You will need this information to modify the `sldap.conf` file. For details, see the following section.

3 Save these changes.

4 Use either the `ldapadd` or `slapadd` command line utility to include a directory entry for this root DN. The `ldapadd` command line is shown below. For detail on how to use these utilities, refer to OpenLDAP’s documentation.

```
ldapadd -x -D “cn=Manager, dc=<mycomany>,dc=<com>” -W -f
<filename>.ldif
```

5 Check that the entry was added correctly with the `ldapsearch` utility. The `ldapsearch` command line is shown below. For details, refer to OpenLDAP’s documentation.

```
Ldapsearch -x -b ‘dc=<mycompany>,dc=<com>’ ‘(objectclass=*)’
```

To make uniqueMember an optional attribute

1. Locate the `core.schema` file in the following OpenLDAP location: `<OpenLDAP_path>/schema/`.
2. Back up the `core.schema` file.
3. Open the `core.schema` file in a text editor.
4. In the `objectclass ( 2.5.6.17 NAME 'groupOfUniqueNames'` section of the file, modify the contents of the `MUST` and `MAY` attributes so it looks like the example provided below.

   ```
   objectclass ( 2.5.6.17 NAME 'groupOfUniqueNames'        DESC
   'RFC2256: a group of unique names (DN and Unique Identifier)'
      SUP top STRUCTURAL
      MUST ( cn )

   MAY ( businessCategory $ seeAlso $ owner $ ou $ o $ description $
   uniqueMember ) )
   ```

5. Save the modified `core.schema` file.

Tuning Directory Servers

The key to proper indexing is knowing what type of searches are most likely to occur. Indexing attributes tunes the performance of Select Access by creating a cache large enough to prevent the Policy Validator from repeated (and unnecessary) disk access.

The Policy Validator frequently evaluates identities with the attributes listed below:

`objectClass`

`userPrincipalName`

`uid`

`memberUid`

`member`

`nxUrl`

`nxUnknownUserRule`

`nxManagement`

`samAccountName`

`cn`

`ou`

`uniqueMember`

`nxRole`

`nxSelectIdRule`

`nxRule`

`nxResource`

and any attributes you have used in dynamic group definitions

HP recommends that you index these attributes before you go into full-scale deployment of Select Access. For information on how to index attributes on your directory server, see your server's documentation.

3 Transparently Supported Web Server Integrations

Of all the web servers Select Access supports, only three integrate transparently with Select Access: Sun/Netscape/iPlanet, IIS web servers, Apache 2.0, and Domino web servers. This chapter describes how the transparent suppport of these web servers enables you to protect web-based resources easily.

Chapter Overview

Table 1 on page 10 lists the web servers Select Access currently integrates with. Topics in this chapter include:

- Integration Overview on page 37
- Preparing Your Web Server for Integration on page 39
- Deploying the Enforcer Plugin on page 39
- Securing Virtual Domains on page 44
- Securing Your Web Server's Resources on page 46
- Denying Access to Suspicious URLs on page 47

Integration Overview

The integration process of a directory server with Select Access is impacted by primarily by the type of web server you are using and whether or not an Enforcer plugin is installed and configured transparently for the server's host computer. Other factors that can affect the deployment of Select Access with your web server include:

- Which authentication mechanism you require
- What level of protection the web servers resources need
- Whether your server performs virtual hosting
- How site data is used with respects to personalization
- Whether you require single sign-on (SSO) across single or multiple domains

Integration Tasks

The following integration tasks are generic to all web servers, whether or not installation and configuration occurs transparently or otherwise.

Task 1: Prepare your Web server for installation.

To make your installation and configuration of your web server and corresponding plugin integrate and operate more effeciently, there are some steps you can consider. For details, see Preparing Your Web Server for Integration on page 39.

Task 2: Integrate the Enforcer plugin with your server.

After you install the corresponding Enforcer plugin for your server, you need to configure the plugin. In some cases, you may need (or want) to manually configure your web server to load the Enforcer plugin. However, in most cases, the Setup Tool manages configuration details automatically. For details, see Deploying the Enforcer Plugin on page 39.

If your web server is a host to virtual domains, there are specific configuration tasks you must perform when you configure your plugin. For details, see Securing Virtual Domains on page 44.

Task 3: Add the server and its resources to the Policy Matrix. Set up authentication and access policy as required.

Determine what you need to do to your web server and its resources to get content protected. The procedure for accomplishing this is not so much contingent upon your server as it is your deployment environment, your security concerns and your business requirements.

For details, see Securing Your Web Server's Resources on page 46.

Task 4: Customize any forms you require for user authentication.

If you plan to perform form-based login, you may want to customize default form templates shipped with Select Access. There are no differences for servers that require manual installation and integration than those that are transparently supported. For details, see Using Forms on Enforcer-protected Web Servers on page 47.

Preparing Your Web Server for Integration

Before you install and configure an Enforcer plugin on your web server consider the steps listed in Table 8.

Table 8 Preparing Your Web Server

Preparation task	Details
1 If your web server is not yet deployed, ensure you have installed it correctly.	• Install your server's software. • Try accessing the main page over HTTP.
2 Deploy other Select Access components.	• Determine what components you or your server requires: — Enforcer plugins for web servers require a running Policy Validator
3 Set up your web content and configure Select Access authentication to control access to top-level content only: — To get faster results because Policy Validator is not being called to authenticate identities against each file — To prevent multiple updates to the access control HTTP cookie — To make maintenance easier because the site is divided into access controlled and non-access controlled areas	• If content is on one server only: — Organize access controlled and non-access controlled files into separate directory trees. — Create an ignored filenames list in the Enforcer plugin's configuration. That way, you do not need to Enforcer-protect security insensitive files. For details, see Chapter 8, Configuring the Enforcer Plugins, in the *HP OpenView Select Access 6.2 Installation Guide*. • If content is on two or more servers: — Set up a separate web server from the one running the Select Access plugin. — Place access controlled content on the server with the Select Access plugin. — Place non-access controlled content on the other server.

Deploying the Enforcer Plugin

The Enforcer plugin is just one component of a scalable Select Access access management solution for your network. For web servers, the Enforcer plugin is the agent for Select Access on the host computer, and acts as the primary point of integration for these servers.

Installing Enforcer plugins for transparently supported web servers is a simple process that is facilitated by a Select Access installer. For details on how to install an Enforcer plugin for these web servers, refer to the *HP OpenView Select Access 6.2 Installation Guide*.

By installing additional Enforcer plugins for each web server on your network, you can effortlessly coordinate Select Access' deployment. With each new installation, Enforcer plugins are registered in the Select Access directory server. Configuration of all components is therefore centralized and easily managed.

Wizard-based Enforcer Configuration and Server Integration

For those Enforcer plugin that support a configuration wizard via the Setup Tool, Select Access ensures the integrity of the system because the Enforcer plugin inherits the configuration stored in the directory at runtime. Updates are made once, thereby reducing the manual effort required and the margin for error that is incurred. In most cases, you do a wizard-based configuration when:

- You want to simplify the configuration of your plugin and transparently manage the integration of the plugin on your server.
- You need to create a bootstrap XML file as part of your integration process.
- Your server's host computer runs on a Windows platform or has X Windows installed on a UNIX platform.

For details on installing and configuring the Enforcer plugin for your web server, see the *HP OpenView Select Access 6.2 Installation Guide*.

Manual Enforcer Configuration and Server Integration

Your web server's configuration file(s) need to be updated with Select Access-specific instructions that load the Enforcer plugin when the server starts. In most cases, you only need to manually modify your web server's configuration file if:

- The Setup Tool fails to modify the necessary web server's corresponding configuration file.
- You want more control over the modification process and would prefer to make the changes yourself.
- You installed an Enforcer plugin in Console mode and cannot use the Setup Tool to integrate the plugin with the web server.

In these cases, you need to know how to manually make the required modifications. For details unique to your web server, see:

- Manually Integrating the Sun/Netscape/iPlanet Enforcer Plugin on page 40
- Manually Integrating the Apache 2 Enforcer plugin on page 43

Manually Integrating the Sun/Netscape/iPlanet Enforcer Plugin

Typically, the Setup Tool configures your web server to load the corresponding Enforcer plugin for it. However, if you need to manually configure your server to load this Sun/Netscape/iPlanet Enforcer plugin, you need to edit your server's corresponding configuration file to load the Enforcer module and several functions:

- The `magnus.conf` file is used to establish a set of global variable settings that affect the server's behavior and configuration.
- The `obj.conf` is used to load the Enforcer module and several functions.

For details, see the appropriate section based on your platform type:

- To load the iPlanet Enforcer plugin for version 6.2 on UNIX on page 41
- To load the Sun/Netscape/iPlanet Enforcer plugin for version 6.2 on Windows on page 41
- To load the Sun/Netscape/iPlanet Enforcer plugin for version 6.2 on UNIX on page 42

To load the iPlanet Enforcer plugin for version 6.2 on UNIX

1. Open /usr/netscape/server4/https-<server name>/config/obj.conf.
2. Locate the `Init fn=load-types mime-types=mime.types` line in `obj.conf`, and add the following lines after it:
    - To load the Sun/Netscape/iPlanet Enforcer plugin module and the functions used by the plugin after this line:

      ```
      Init fn=”load-modules” shlib=<install_path>/bin/<module>
      funcs=  “enforcer_init,enforcer_check,enforcer_content”
      ```

      Type this line all on one line. If you include any line breaks in it, the web server fails to start and generates an error message.

      where `<module>` is either:

      — `iplanet_web.so` on Linux and Solaris

      — `iplanet_web.sl` on HP–UX
    - To initialize the Sun/Netscape/iPlanet Enforcer plugin at server startup:

      ```
      Init fn=enforcer_init
      ```
3. Locate the `<Object name=default>` section, and add the following lines to it:
    - To check the path:

      ```
      PathCheck fn=”enforcer_check”
      ```
    - To display dynamic content to end users:

      ```
      Error fn=enforcer_content reason=”ENFORCER_CONTENT”
      ```
4. Open `magnus.conf`.
5. Change the value of the `StackSize` parameter to `393216`. This prevents fatal errors from occurring in the Sun/Netscape/iPlanet Enforcer plugin. If the default value is used, the Sun/Netscape/iPlanet Enforcer plugin runs out of stack.
6. Restart your web server.

To load the Sun/Netscape/iPlanet Enforcer plugin for version 6.2 on Windows

1. Open `<iP_install_path>\iPlanet\servers\https-<server name>\config\obj.conf`.
2. To perform a path check, locate the `NameTrans fn=document-root root=”$docroot”` line in `obj.conf`, and add the following lines after it:

    ```
    PathCheck fn=”enforcer_check”
    ```
3. To display dynamic content to end users, locate the `AddLog fn=flex-log name=”access”` line, and add the following line after it:

    ```
    Error fn=”enforcer_content” reason=”ENFORCER_CONTENT”
    ```
4. Open `magnus.conf`.
5. Locate the `Init fn=load-types mime-types=mime.types` line, and add the following lines after it:
    - To load the Sun/Netscape/iPlanet Enforcer plugin module and the functions used by the plugin:

```
Init fn=”load-modules” shlib=<install_path>\bin\iplanet_web32.dll
funcs=“enforcer_init,enforcer_check,enforcer_content”
```

> Type all of this on one line. If you include any line breaks in it, the web server fails to start and generates an error message.

> While the Enforcer plugin’s name has changed to Sun ONE, the module file name has not. Enter the line exactly as it has been defined in the example above.

- To initialize the Sun/Netscape/iPlanet Enforcer plugin when the server starts:

  ```
  Init fn=enforcer_init
  ```

6. Change the value of the `StackSize` parameter to `393216`. This prevents fatal errors from occurring in the Sun/Netscape/iPlanet Enforcer plugin. If the default value is used, the Sun/Netscape/iPlanet Enforcer plugin runs out of stack.
7. Restart your web server.

To load the Sun/Netscape/iPlanet Enforcer plugin for version 6.2 on UNIX

1. Open `/usr/iplanet/servers/https-<server name>.com/config/obj.conf`.
2. To perform a path check, locate the `NameTrans fn=document-root root=”$docroot”` line in `obj.conf`, and add the following line after it:

   ```
   PathCheck fn=”enforcer_check”
   ```

   This line is a `PathCheck` directive.
3. To display dynamic content to end users, locate the `AddLog fn=flex-log name=”access”` line, and add the following line below it:

   ```
   Error fn=”enforcer_content” reason=”ENFORCER_CONTENT”
   ```
4. Open `magnus.conf`.
5. Locate the `Init fn=load-types mime-types=mime.types` line, and add the following lines:

   - To load the Sun/Netscape/iPlanet Enforcer plugin module and the functions used by the plugin:

     ```
     Init fn=”load-modules” shlib=<install_path>/bin/<module>
     funcs= “enforcer_init,enforcer_check,enforcer_content”
     ```

     > Type all of this on one line. If you include any line breaks in it, the web server fails to start and generates an error message.

   where *<module>* is either:

   — `iplanet_web.so` on Linux and Solaris

   — `iplanet_web.sl` on HP–UX.

   > While the Enforcer plugin’s name has changed to Sun ONE, the module file name has not. Enter the line exactly as it has been defined in the example above.

   - To initialize the Sun/Netscape/iPlanet Enforcer plugin when the server starts:

     ```
     Init fn=enforcer_init
     ```

6. Change the value of the `StackSize` parameter to `393216`. This prevents fatal errors from occurring in the Sun/Netscape/iPlanet Enforcer plugin. If the default value is used, the Sun/Netscape/iPlanet Enforcer plugin runs out of stack.

7. Restart your web server.

Manually Integrating the Apache 2 Enforcer plugin

Typically, the Setup Tool configures your web server to load the corresponding Enforcer plugin for it. However, if you need to manually configure your server to load the Apache 2 Enforcer plugin, edit your `httpd.conf` file to load the Enforcer module and several functions. Depending on your deployment, see the corresponding section:

- To edit the httpd.conf file on UNIX on page 43
- To edit the httpd.conf file on Windows on page 44

To edit the httpd.conf file on UNIX

1. Make sure the Policy Validator is running.

2. Open your `httpd.conf` Apache configuration file:

   ```
   vi /etc/httpd/conf/httpd.conf
   ```

3. Add the following lines to the beginning of the `LoadModule` section. These lines ensure that the Apache 2 Enforcer plugin starts after the SSL module has already loaded. Depending on your deployment of the Apache server, this location varies:

   - If Apache uses its own access control handler, place the lines above the `enforcer_module` line in the `LoadModule` section.
   - Otherwise, add these lines after the `mod_ssl` section. Use either of the following platform-specific lines, as appropriate for your host computer:
     — For HP–UX:

     ```
     AddModule enforcer_module <install_path>/bin/mod_enforcer.cpp
     LoadModule enforcer_module <install_path>/bin/mod_enforcer.sl
     ```

     — For Linux and Solaris:

     ```
     AddModule enforcer_module <install_path>/bin/mod_enforcer.cpp
     LoadModule enforcer_module <install_path>/bin/mod_enforcer.so
     ```

4. If you use SSL, add the following line to export SSL data the Policy Validator requires. Without adding this line, any SSL encryption decision points in your existing rules fail.

   ```
   SSLOptions +ExportCertData +CompatEnvVars +StdEnvVars
   ```

5. Comment out the `ClearModuleList` line.

6. Save `httpd.conf` and restart the Apache web server. If you are running the Apache web server on HP-UX, you need to start the web server manually or set `LD_PRELOAD` in the environment. For details, refer to Chapter 8, Configuring the Enforcer Plugins, in the *HP OpenView Select Access 6.2 Installation Guide*.

To edit the httpd.conf file on Windows

1. Make sure the Policy Validator is running.
2. Open your `<WS_install_path>\conf\httpd.conf` file:
3. Add the following lines to the beginning of the `LoadModule` section.

   ```
   LoadModule enforcer_module "<install_path>\bin\apache_web32.dll"
   AddModule enforcer_module "<install_path>\bin\mod_enforcer.cpp"
   ```

   Depending on your deployment of the Apache server, this location varies:

   - If Apache uses its own access control handler, place the lines above the `enforcer_module` line in the `LoadModule` section.
   - Otherwise, add these lines after the `mod_ssl` section. This ensures that the Apache 2 Enforcer plugin starts after the SSL module has already loaded.
4. Save `httpd.conf` and restart the IBM HTTP server distributed with WebSphere.

Securing Virtual Domains

Unlike a dedicated server that has been set up to service a single domain, the hosting server of virtual domains can service multiple domains all from the same machine.Virtual domains are web sites that are hosted by a third party. Typically, sites that do not need high scalability or high bandwidth use virtual domains. Virtual domains are typically set up as host names or IP addresses. However, if you are using either the Sun ONE server, the IIS server, or the Apache server, virtual domains can also be set up as headers.

If you experience problems configuring virtual web server support on IIS while running on Windows 2000 with Service Pack 2, there are a few ways to resolve the problem. For details, see Appendix E, Troubleshooting, in the *HP OpenView Select Access 6.2 Policy Builder Guide*.

Creating a Pass-through Domains List

Depending on the specific hosting environment, host organizations may not want to incur the cost on performance that occurs when you Select Access-protect a web server and consequently all virtual domains. To get host environments to selectively choose which domains to Select Access-protect, you can create a **Pass-through Domains** list. You configure this list when you set up an Enforcer plugin for the web server that is acting as a host to these virtual domains. For details on how to configure a list, see Chapter 8, Configuring the Enforcer Plugins, in the *HP OpenView Select Access 6.2 Installation Guide*.

Creating an unprotected domains list requires that you choose to do a **Custom** setup. You select this option from the Enforcer plugin's **General** setup screen in the Setup Tool.

Scenario: Virtually Yours hosting service

Virtually Yours is a hosting service that has a web server called VirtualIIS. This host machine has different network cards for each of the following addresses with text names that have been registered with InterNIC.

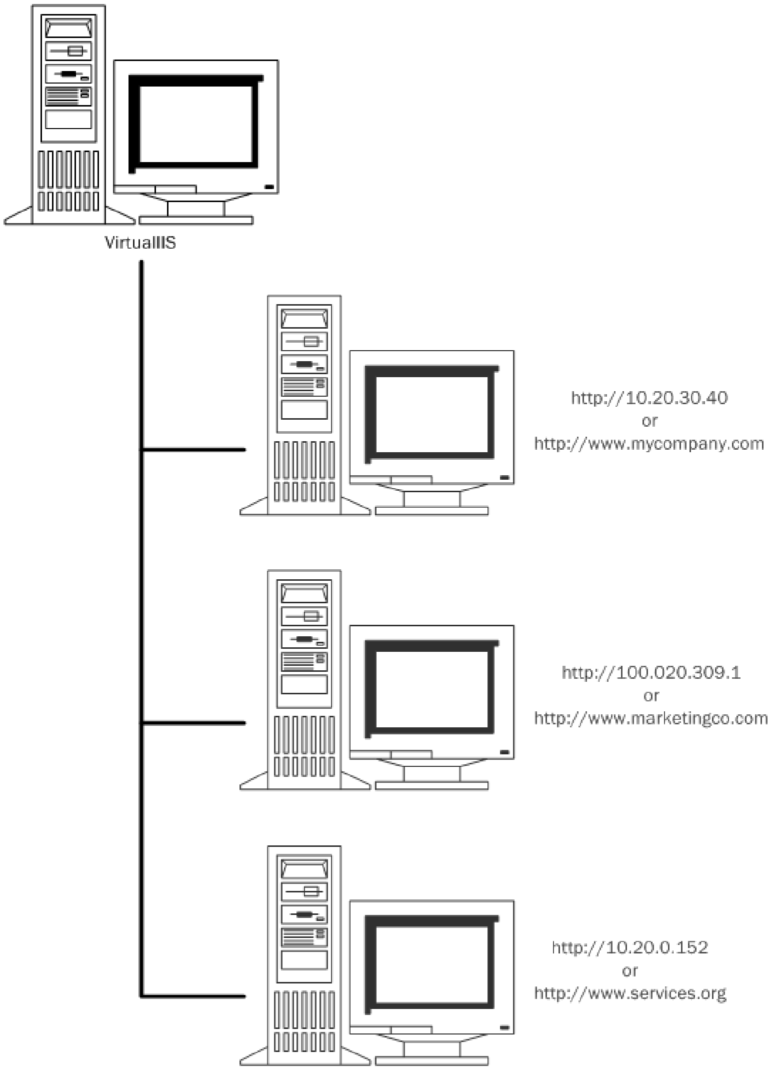


Figure 5 Web Server with Virtual Domains

Consider the following:

- `mycompany.com` is VirtuallyYours' largest client (to whom they have charged an extra fee for additional security services).
- `marketingco.com` and `services.org` are small bandwidth sites that have bought VirtuallyYours' basic package.

Then, Virtually Yours might configure VirtualIIS' IIS Enforcer plugin with a **Pass-through Domain** list that might only include the following names:

```
marketingco
services
```

Securing Your Web Server's Resources

Once you have successfully integrated the Enforcer plugin on your web server's host computer, you can begin securing your resources. The Policy Matrix provides a clear view of all users and resources. It lets you understand the overall policy for specific user and resource combinations, which allows you to answer questions like "Who can access a specific resource?" or "Which resources does an individual have access to?".

Discovering Your Resources

If you have setup your servers as described in Preparing Your Web Server for Integration on page 39, building a tree of resources you want to secure is simplified. To deploy Select Access even more quickly, the Policy Builder includes an innovative scanning mechanism called a Automatic Resource Discovery, that automatically enumerates all network services and resources, such as URLs, dynamic pages and portal links. The results are displayed in the Policy Matrix and added to the directory in a hierarchical format that best reflects intranet, extranet, or portal resources. For explicit details on this tool, refer to Chapter 4, Organizing Identities and Resources, in the *HP OpenView Select Access 6.2 Policy Builder Guide*.

To secure your resources

1. Add the web server to the Resources Tree in the Policy Matrix, defining the protocol as either HTTP or HTTPS (for SSL). For details, see Chapter 3, Building your Identities and Resources Trees in the *HP OpenView Select Access 6.2 Policy Builder Guide*.
2. Set the access policy for this web server. Pairings of identities, resources and associated policy rules are displayed with policy rule icon: "allow," "deny" or "conditional". Policy inheritance allows you to automatically set policy for groups of identities and resources, thereby eliminating the need to manually assign policy for every identity and/or resource.

   Good policy setting practice dictates that you always determine identity access with a policy and not an authentication method. When you try to use the authentication service to determine access logic, the final access decision can be unpredictable.

3. Choose the authentication methods used to identify unknown users. Select Access supports multiple authentication types, including user self-registration and passwords, Windows NT, Kerberos, SecurID, tokens and X.509 certificates.
4. Access the web server using HTTP to confirm that the Policy Enforcer plugin is installed and configured correctly. Verify that access is allowed or denied based on the policy you set and the authentication method you have defined.

   Once you have set up the web server following these instructions, no one can access resources on this web server unless they first:

   - Authenticate themselves as a known user in the directory server
   - Have an access policy that gives them access to a protected resource

Denying Access to Suspicious URLs

The Enforcer plugin considers the following URLs as suspicious:

- If the path requested by a user contains the substrings "/..", "/./", or "//".
- If the path requested by a user looks like an 8.3 truncation (for example, certain HTML editors truncate `abcdefghijklmnop.html` to `abcdef~1.htm`).

➤ The only time the Enforcer plugin does not consider a URL with a tilde (~) character as suspicious, is when the tilde is the first character position. For example:

```
<protocol>://<URL_path>/~myhomepage
```

In these cases, the Enforcer plugin denies access immediately. In the latter case, because of the dangers associated with 8.3 filename remapping.

To determine whether or not your web content is affected by suspicious URLs:

1. Open logs generated by the Enforcer plugin in the location it has been outputted to.
2. Search your logs for the following string:

```
"rejecting suspicious url <actual_URL_requested>"
```

Using Forms on Enforcer-protected Web Servers

To give you the ability to use and manage logins and profiles, Select Access uses a system of HTML forms to support its internal authentication, authorization, and profile mechanisms:

- Login forms are used when security administrators configure a matching authentication service. For configuration details, see Chapter 6, Setting Up Authentication Services, in the *HP OpenView Select Access 6.2 Policy Builder Guide*.
- Self-management forms (password and profile) are used when security administrators define the directory attributes that can be self-managed by the individual end users (e.g. addresses, phone numbers, office location), and/or explicitly set password management as a identity-based subset of the enterprise password policy. For configuration details, see Chapter 9, Managing Identity Profiles, in the *HP OpenView Select Access 6.2 Policy Builder Guide*.

Using Forms for Form-based Login

Form-based login is one of the four known web-based login mechanisms used by web servers today. It is typically used in conjunction with or as an alternative to other login alternatives like:

- HTTP basic authentication
- HTTP digest authentication
- HTTPS mutual authentication

!! If you intend to use HTTP basic authentication, disable form-based login to avoid conflicts that can occur. For details, see HTTP Basic Authentication Problematic on page 323 in the *HP OpenView Select Access 6.2 Policy Builder Guide*.

Major advantages of using Select Access' form-based login are:

- You can customize the look and feel of the login page, especially in contrast to the HTTP browsers' built-in mechanisms.
- You prevent the information from being cached in the web browser. This in turn:
    - Allows the Policy Validator to terminate a user session
    - Prevents unauthenticated identities from gaining access to information and resources via browsers that have been left open

Using Forms for Profile Self-Management

Keeping identity data timely requires that you minimize the number of intermediaries that maintain identity data. Not only does this minimize corporate maintenance expenses, but it also allows end users to implement immediate changes. To that end, Select Access includes a rich set of native password and profile management capabilities that mitigate administrative expenses by allowing end users to directly manage the elements in their profile that help define the identity of the individual.

A profile consists of two things:

- A set of activated attributes that create a profile for an identity. If security administrators create a rule with the self-management terminal point, they can determine which of these attributes the end user can self-manage.
- A profile-specific subset of the corporate password policy that sets unique self-management preferences for the identity.

Using Forms with Your Web Server

Select Access includes several forms that are used to support Select Access' key features. By default, these forms are installed in the `<install_path>/content` folder. To use forms with your web server, consider the steps listed in Table 9. The types of forms listed in Table 10 categorize the types of forms you can use with your web server.

⚠ You cannot use non-Select Access forms to collect POST data. Select Access Enforcer plugins expect data in specific places. Other forms do not capture and forward data as the Enforcer plugins may expect it.

Table 9 Configuring Your System to use Select Access Forms

Setup task	Details
1 Do one of the following: — If this is your first installation: Modify Select Access' form templates to suit your business needs. — If your are upgrading: Modify the new forms shipped with Select Access. Use your old forms for reference purposes only. New forms can often contain new parameters that are vital to the version of Select Access you have just installed.	Customizing Select Access Forms and Messages on page 50 OR Upgrading Forms to Ensure Correct Behavior on page 58
2 If you want your login forms to display why authentication has failed, customize Select Access behavior as required.	Giving Reasons for Authentication Failures on page 59 AND Customizing Forms to Use Authentication Failure Reasons on page 60
3 Save your files to the `<install_path>/content` directory on all Enforcer-protected web servers. — You cannot change the location of this directory, otherwise the Enforcer plugin does not know where to locate these forms. — You can only change the names of forms. For all other message files like `accepted.html` and `deny.html`, you cannot change the filename. However, if you change a form's name, remember to update the corresponding the authentication service's form properties.[a]	Chapter 6, Setting Up Authentication Services, in the *HP OpenView Select Access 6.2 Policy Builder Guide*
4 Configure your Select Access-protected system to preserve POST data for resources where access is controlled.	Preserving POST Data on page 61

[a] Message page names are hard-coded into the Enforcer plugins. If you alter the name in any way, Enforcer-protected web servers cannot display these pages to the end user.

Customizing Select Access Forms and Messages

You can change any part of Select Access forms, except parameters that are classified as “restricted”. Restricted parameters are those elements of the form templates that Select Access requires for internal functions. If you modify or delete any of these parameters, unexpected results can occur when your end users use these forms. Table 10 list the forms you can customize.

Table 10 Available Form Templates

Default Filename	Description	Restricted Elements
`accepted.html`	Notifies the end user that their credentials are accepted and that their resource request is pending.	none
`deny.html`	For IIS Enforcer plugins only, notifies the end user that their credentials have been denied. IIS web servers do not have a default deny page.	none
`login_form.html`	Collects usernames and passwords as an alternative to HTTP basic authentication.	• Form fields • Submit button • Hidden refresh parameters For details, see To customize the Login form on page 52.
`password_dictionary_match_form.html`	Notifies end users that passwords match words defined at setup time in the Policy Validator password dictionary.	none
`password_expired_form.html` `password_expiry_form.html`	Notifies end users that their password is about to expire and gives them the option of: • Changing it now. • Changing it at a more suitable time. If the end user chooses to change their password now, the Enforcer plugin displays `password_change_form.html`.	The order and name of the two submit buttons. For details, see To customize the Password Expiry form on page 53.
`password_history_match_form.html`	Notifies end users that the password they provided matches one in their password history.	none

Table 10 Available Form Templates (cont'd)

Default Filename	Description	Restricted Elements
password_invalid_length_form.html	Notifies end users that their password does not meet the minimum and/or maximum password lengths.	none
password_missing_chars_form.html	Notifies end users that their password does not meet the uniqueness characters required by the system.	none
password_username_match_form.html	Notifies end users that passwords cannot include their given name, family name, UID, or CN.	none
password_change_form.html	Presents the fields required to change passwords.	• Form fields • Submit button • Hidden form attribute For details, see To customize the Password Change form on page 53.
registration_form.html	Allows end users to register and creates a profile on the directory server. This allows administrators to set explicit policies for them, rather than just inheriting the unknown identity policy.	• Form fields • Submit button • Hidden refresh parameters For details, see To customize the Registration form on page 54.
multiauth_form.html	Collects multiple authentication inputs from the end user when two or more authentication mechanisms are required to access a resource (part of SelectAuth).	Variable, depending on the authentications you want to include. For details, see To customize the Multiple Authentication Services form on page 54.
radius_form.html	Presents challenges to and captures their responses from end users, which the Enforcer plugin forwards to the RADIUS server via the Policy Validator.	Form fields only. For details, see To customize the RADIUS form on page 56.

Table 10 Available Form Templates (cont'd)

Default Filename	Description	Restricted Elements
`winauth_form.html`	For IIS Enforcer plugins only, displays a login form that supports NTLM and Kerberos authentication. Like the password login form, this form collects credentials (username, password, and domain name) on a separate form, in order to prevent the information from being cached in the web browser.	• Form fields • Submit button • Hidden refresh parameters • Hidden form attribute For details, see To customize the Integrated Windows (NTLM/Kerberos) form on page 56.
`profile_mgmt_form.html`	Notifies end users to update their profile.	• Form fields required by Select Access only • Submit button For details, see To customize the Profile Self-Management form on page 57.
`password_change_ form.html`	Notifies end users to update their password.	none
`profile_error_ form.html`	Notifies end users of a profile update error. For example, this error form appears with failed connection between the Policy Validator and the directory server.	none
`profile_no_user_ form.html`	Notifies end users that they lack a profile on directory server.	none
`profile_ok_ form.html`	Notifies end users when their profile is successfully updated.	none

To customize the Login form

1. Add or modify text and graphics as required.
2. Do not change the order, names, or values of the following form elements:
   - The first field in the form must be:

     ```
     <INPUT TYPE="text" NAME="user" VALUE="">
     ```

   - The second field must be:

     ```
     <INPUT TYPE="password" NAME="password" VALUE="">
     ```

3. If required, you change the location or value of the submit element only:

   ```
   <INPUT TYPE="submit" NAME=".submit" value="Login now">
   ```

4. Ensure the following refresh parameter appears somewhere on this page.

   ```
   <INPUT TYPE-”HIDDEN” NAME=”selectaccess_send_refresh”
   VALUE=”enable”/>
   ```

   ➤ This field is required by Enforcer-protected web servers, and causes the web browser to perform a GET, rather than a POST. Ensure you understand the implication of this line and plan your web site content accordingly.

5. If you do not want your identities to know what password server or realm they are authenticating against, you can delete the following line:

   ```
   Password Server: %%form_realm%%
   ```

To customize the Password Change form

1. Add or modify text and graphics as required.
2. Do not change the order, names, or values of the following form elements:
   - The first field must be:

     ```
     <INPUT TYPE="text" NAME="user" VALUE="%%user%%">
     ```

   - The second field must be:

     ```
     <INPUT TYPE="password" NAME="oldpassword" VALUE="">
     ```

   - The third field must be:

     ```
     <INPUT TYPE="password" NAME="newpassword1" VALUE="">
     ```

   - The fourth field must be:

     ```
     <INPUT TYPE="password" NAME="newpassword2" VALUE="">
     ```

3. If required, you can change the location or value of the submit element only:

   ```
   <INPUT TYPE="submit" NAME="password_mgmt" value="Change Password">
   ```

4. Make sure the last line in the form contains the hidden attributes required by Select Access:

   ```
   <INPUT TYPE="HIDDEN" NAME="form_realm" VALUE="%%form_realm%%"/>
   ```

To customize the Password Expiry form

1. Add or modify text and graphics as required.
2. Do not modify the `%%days_remaining%%` text substitution placeholder. It is used to dynamically insert the correct number of days remaining before the end user’s password expires.
3. Do not change the order or names of the following form elements:
   - The first field must be:

     ```
     <INPUT TYPE="submit" NAME="password_change" value="Change
     Password Now">
     ```

   - The second field:

     ```
     <INPUT TYPE="submit" NAME="password_defer" value="Change
     Password Later">
     ```

To customize the Registration form

1. Add or modify text and graphics as required.
2. Do not change the order, names, or values of the following form elements:
   - The first field must be:

     ```
     <INPUT TYPE="text" NAME="givenName" VALUE="">
     ```

   - The second field must be:

     ```
     <INPUT TYPE="text" NAME="sn" VALUE="">
     ```

3. If your directory requires other mandatory profile attributes, or you have activated other attributes for an identity, add them below the second field. Field elements use the following syntax:

   ```
   <INPUT TYPE="text" NAME="<attribute_name>" VALUE="">
   ```

   where *<attribute_name>* is the name of the directory attribute required by your server.

   For example, eTrust directory servers require fax and telephone number fields. In this instance, you would add the following lines:

   ```
   <INPUT TYPE="text" NAME="telephonenumber"
   VALUE="%%telephonenumber%%">
   ```

   ```
   <INPUT TYPE="text" NAME="facsimiletelephonenumber" VALUE="">
   ```

4. Add asterisks or some other convention to indicate the fields that require mandatory input from the identity.
5. If required, you can change the location or value of the submit element only:

   ```
   <INPUT TYPE="submit" NAME=".submit" value="Register now">
   ```

6. In the already registered section, you can change the location or value of the submit element only:

   ```
   <INPUT TYPE="submit" NAME=".login" value="Login now">
   ```

7. Make sure you include the following field somewhere on this page:

   ```
   <INPUT TYPE=”HIDDEN” NAME=”selectaccess_send_refresh”
   VALUE=”enable”/>
   ```

   ➤ This field is required by Enforcer-protected web servers, and causes the web browser to perform a GET, rather than a POST. Ensure you understand the implication of this line and plan your web site content accordingly.

   ⚠ If you are using the Administration server to register your end users as Select Access identities, ensure the JSP form collects the person’s first name. Otherwise, end users are able to create passwords that include part of their name or user ID.

To customize the Multiple Authentication Services form

1. Add or modify text and graphics as required.

   For example, because no message is displayed when an authenticated, known identity requests a denied resource deny policy is set, end users might think they had incorrectly entered their credentials. Therefore, you may want to add a message that notifies end users of this possibility.

2 Add, delete, and/or reorder sections of the form-based on the authentication methods your company uses, including any custom authentication plugins you have written and uploaded.

3 Within sections for each authentication method:

- Reorder, comment out, and/or add fields.
- However, if you require password, SecurID, and/or RADIUS authentication, then do not change the order, names, or values of the following form elements:
  - The first field must be:

    ```
    <INPUT TYPE="text" NAME="user" VALUE="%%user%%">
    ```

  - The second field must be:

    ```
    <INPUT TYPE="password" NAME="password" VALUE="">
    ```

- Additionally, if you require registration authentication, do not change the names or values of the following form elements:
  - The **Name** field must be:

    ```
       <INPUT TYPE="text" NAME="givenname" VALUE="">
    ```

  - The **Last name** field must be:

    ```
       <INPUT TYPE="text" NAME="sn" VALUE="">
    ```

  - The **Organization name** field must be:

    ```
       <INPUT TYPE="text" NAME="organizationname" VALUE="">
    ```

  - The **Email address** field must be:

    ```
       <INPUT TYPE="text" NAME="mail" VALUE="%%mail%%">
    ```

  - The **Phone number** field must be:

    ```
       <INPUT TYPE="text" NAME="telephonenumber"
    VALUE="%%telephonenumber%%">
    ```

  - The **Fax number** field must be:

    ```
       <INPUT TYPE="text" NAME="facsimiletelephonenumber" VALUE="">
    ```

- If your directory requires other mandatory profile attributes, or you have activated other attributes for an identity, add them below the second field. Field elements use the following syntax:

  ```
  <INPUT TYPE="text" NAME="<attribute_name>" VALUE="">
  ```

  where *<attribute_name>* is the name of the directory attribute required by your server.

- Add asterisks or some other convention to indicate the fields that require mandatory input from the end user.
- If required, you can change the location or value of the submit element only:

  ```
  <INPUT TYPE="submit" NAME=".submit" value="Register now">
  ```

To customize the RADIUS form

1. Add or modify text and graphics as required.
2. Do not change the hidden attribute for the form element. For example:

   ```
   <INPUT TYPE-”HIDDEN” NAME=”form_radius_state”
   VALUE=”%%form_radius_state%%”/>
   ```

3. Make sure you include the following field somewhere on this page:

   ```
   <INPUT TYPE=”HIDDEN” NAME=”selectaccess_send_refresh”
   VALUE=”enable”/>
   ```

   This field is required by Enforcer-protected web servers, and causes the web browser to perform a GET, rather than a POST. Ensure you understand the implication of this line and plan your web site content accordingly.

4. If you do not want your end users to know what password server or realm they are authenticating against, you can delete the following line:

   ```
   Authentication Server: %%form_realm%%
   ```

To customize the Integrated Windows (NTLM/Kerberos) form

1. Add or modify text and graphics as required.
2. Do not change the order, names, or values of the following form elements:
   - The first field must be:

     ```
     <INPUT TYPE="text" NAME="user" VALUE="">
     ```

   - The second field must be:

     ```
     <INPUT TYPE="password" NAME="password" VALUE="">
     ```

   - The third field must be:

     ```
     <INPUT TYPE="text" NAME="domain" VALUE="">
     ```

3. If required, you can change the location or value of the submit element only:

   ```
   <INPUT TYPE="submit" NAME=".login" value="Login now">
   ```

4. Make sure you include the following field somewhere on this page:

   ```
   <INPUT TYPE=”HIDDEN” NAME=”selectaccess_send_refresh”
   VALUE=”enable”/>
   ```

   This field is required by Enforcer-protected web servers, and causes the web browser to perform a GET, rather than a POST. Ensure you understand the implication of this line and plan your web site content accordingly.

5. If you do not want your end users to know what password server or realm they are authenticating against, you can delete the following line:

   ```
   %%sspi_package%% Server: %%form_realm%%
   ```

To customize the Profile Self-Management form

1. Add or modify text and graphics as required.
2. Add or comment out form fields as required.

   - If you did not activate the recommended attribute, comment out the default field.
   - If you activated an attribute (other than a recommended attribute), add a field to the form.

   For example, if you have activated additional attributes and need to add one or more form fields, the format for adding fields is the following:

   ```
   <INPUT TYPE=”text” NAME=”attribute_name” VALUE=”%%value_type%%”>
   ```

   where,

   — `TYPE` is always `”text”` because end users only input ASCII data.

   — `NAME=”attribute_name”` is the name of the additional attribute you activated.

   — `VALUE=”%%value_type%%”` is the type of the attribute. Insert the details between the % characters. The Enforcer plugin looks for the data between these characters and replaces the information in LDAP.

   Note that you cannot modify these elements because Select Access requires them:

   - The **First name** field must be: `<INPUT TYPE="text" NAME="givenname" VALUE="%%givenname%%">`
   - The **Last name** field must be:

     ```
     <INPUT TYPE="text" NAME="sn" VALUE="%%sn%%">
     ```

   - The **Email address** field must be:

     ```
     <INPUT TYPE="text" NAME="mail" VALUE="%%mail%%">
     ```

   - The **Phone number** field must be:

     ```
     <INPUT TYPE="text" NAME="telephonenumber"
     VALUE="%%telephonenumber%%">
     ```

   - The **Fax number** field must be:

     ```
     <INPUT TYPE="text" NAME="facsimiletelephonenumber"
     VALUE="%%facsimiletelephonenumber%%">
     ```

3. If you have activated the corresponding password attribute for your directory server, you may need to change the name of the attribute for the change password element, if the one provided in the template is incorrect (for example, `password_id` versus `userPassword`).
4. If you do not want end users to change their own password:

   - Deactivate the corresponding password attribute for your directory server
   - Comment out this button. Because the password attribute varies depending on which directory server you use

Upgrading Forms to Ensure Correct Behavior

When updating from a previous version to Select Access 6.2, you must ensure that you update customized forms used for form-based logins. Forms you need to manually update to get this correct behavior include:

- `login_form.html`
- `multiauth_form.html`
- `password_change_form.html`
- `password_change_unable_form.html`
- `password_confirm_mismatch_form.html`
- `password_dictionary_match_form.html`
- `password_expired_form.html`
- `password_expiry_warning_form.html`
- `password_history_match_form.html`
- `password_invalid_length_form.html`
- `password_missing_chars_form.html`
- `password_missing_field_form.html`
- `password_userid_notfound_form.html`
- `password_username_match_form.html`
- `radius_form.html`
- `registration_form.html`

By default, the installer backs up pre-existing forms you have modified to a subfolder of the `<install_path>/content/` folder. Customized forms from previous versions are missing:

- A hidden refresh parameter
- A hidden submit parameter

These parameters are critical to the customized forms' behaving correctly with this patched version. Otherwise, when a form is filled in and then submitted, the end user cannot be redirected to the intended page. Instead, an error message is displayed.

To update your custom forms

1. Open a customized form.
2. Add the following:
   - If you are using an IIS v5.0 web server, add the submit parameter anywhere between the `<FORM>` and `</FORM>` tags:

     ```
     <INPUT TYPE="HIDDEN" NAME="SA_PRESERVED_POST"
     VALUE="%%SA_PRESERVED_POST%%">
     ```

   - For all servers, add the hidden refresh parameter anywhere between the `<FORM>` and `</FORM>` tags:

     ```
     <INPUT TYPE="HIDDEN" NAME="selectaccess_send_refresh"
     VALUE="enable"/>
     ```

3. Save this file to the `<install_path>\content` folder.

   ➤ If you do not overwrite the new file in this location, the Enforcer plugin does not know to use your custom file. Instead, the Enforcer plugin uses the default file installed with this patch to collect authentication information from the identity.

4. Repeat Steps 1-3 for each form you have customized.

   ➤ If you have customized the `multiauth_form.html` file or `registration_form.html`, ensure you add the refresh and submit parameters for each occurrence of `<FORM>` and `</FORM>` tags.

5. When you have overwritten all the files in the `<install_path>\content` folder, delete the subfolder.

Giving Reasons for Authentication Failures

The Policy Validator replies include a parameter to describe authentication failures: `authfail_reason`. This new parameter can have one of the following values:

- `Authentication failed`
- `Account disabled`
- `Account disabled due to too many invalid passwords`
- `Account disabled due to inactivity`
- `Unknown user name`
- `Incorrect password`
- `Logged out from session`
- `Session timed out`
- `User name already exists`
- `Missing information`
- `Password must be changed`
- `Password expired`
- `Password will expire soon`
- `Password change attempt failed`
- `New password does not meet rules`

Displaying Authentication Failure Details to End-users

While the Policy Validator automatically provides this information to the Enforcer plugin, this information does not get passed on to end users by default. Should you want to display this information for the individual, you must customize the forms in the content directory in order to pass this information on to the end user.

Forwarding authentication failure details can be useful for the end user. For example, this information could provide greater assistance to end users to help them correct the error that led to the failure, thus reducing the amount of help desk support needed. In addition, with this information, web applications could be developed which respond to authentication failures differently based on the reason for the failure.

⚠ Before customizing forms to make authentication failure information available to end users, you should be aware of the security implications that may arise in doing so. For more information, see Benefits & Security Implications on page 60.

Benefits & Security Implications

Explaining and describing authentication failure reasons can:

- Provide some usability benefits to the end user
- Lessen the need for help desk support

However, it also represents a potential security concern for the organization: the more an attacker knows about why authentication failed, the better able the attacker is to determine how security might be breached. Before passing authentication failure information on to end users, we recommend that you thoroughly assess the risks and weigh them against the possible benefits.

Customizing Forms to Use Authentication Failure Reasons

You can add the parameter directly to the forms to display the reason for an authentication failure, or you can use it in JavaScripts to localize the messages or provide some additional logic to control how the message is displayed to the end user. To see an example of how JavaScript can be used to localize the message, see Using JavaScript to Localize Authentication Failure Messages on page 60.

To customize forms

1. Open one of the affected login or registration forms. The forms can be found in the `<install_path>\content` folder.
2. Add the following parameter anywhere between the `<FORM>` and `</FORM>` tags:

   ```
   %%authfail_reason%%
   ```

3. Save the file.
4. Repeat Steps 1-3 for each form as needed.

Using JavaScript to Localize Authentication Failure Messages

Using JavaScript, you can create simple scripts to localize the authentication failure message. The simplest way to do this is to use a `switch` statement. The code sample below provides a sample localization script.

Sample JavaScript

```
<SCRIPT LANGUAGE="JavaScript">
  switch ("%%authfail_reason%%") {
    case "Authentication Failed":
      document.write("Authentisierung Fiel");
```

```
   break;
  case "Account Disabled":
   document.write("Account Sperrte");
   break;
  default:
   document.write("%%authfail_reason%%");
 }
</SCRIPT>
```

Preserving POST Data

The Enforcer plugin preserves POSTed data when the destination of that POST is a protected resource to which access is restricted. When the Enforcer plugin requires authentication with Select Access forms, information is not lost as the identity is prompted for authentication.

You cannot use non-Select Access forms to collect POST data. Select Access Enforcer plugin expect data in specific places. Other forms do not capture and forward data as the Enforcer plugins may expect it.

The Enforcer plugin evaluates the initial request to see if the request was a POST. If so, the Enforcer plugin dynamically generates a form that re-POSTs the original form data included in hidden fields in one of two ways:

- If JavaScript is enabled in the browser, the Enforcer plugin triggers the POST with JavaScript. The re-POSTing of form data occurs so quickly that it is virtually transparent to the end user.
- If JavaScript is disabled in the browser, the Enforcer plugin displays the form with a Submit button, which allows the end user to manually trigger it.

  If you use the greater than (>) and less than (<) characters in your redirect URLs, you need to re-write links so that these characters are properly URL-encoded. This URL encoding prevents URLs from being misinterpreted as cross-site scripting attempts.

  - Use the `&gt;` escape characters to URL encode the greater than (>) character.
  - Use the `&lt;` escape characters to URL encode the less than (<) character.

  Manually triggering a re-POST is more apparent on a multidomain single sign-on (MD-SSO) deployment of Select Access. In this case, the end user will need to re-submit the form data for each redirect that the Enforcer plugin requires for SSO authentication.

To testing the plugin's POSTing ability

If you test the Enforcer plugin's new POSTing ability with an IE browser on the same host machine as one of your MD-SSO servers, POST data is always lost due to an incompatibility in the Enforcer plugin with IE browser and IIS v5.0 servers. You can correctly test this POST functionality when you:

- Test the POST functionality from a different machine other than the host computer of your web servers.

  OR

- Use a Netscape browser.

Using Forms to POST with International Characters

If you intend to use forms to POST data that includes international characters *before* the identity has been authenticated, you must take the necessary precautions, otherwise the data becomes distorted. To avoid the distortion of non-ASCII characters, HP recommends that you always ensure that the identity has been authenticated to avoid the problematic POST-over functions.

Distortion only occurs with C/C++ Enforcer plugins. These plugins trigger either POST-over-auth or POST-over-MDSSO in order to authenticate the identity; the Enforcer plugin cannot restore the initial POST request so that the web application can continue processing

C/C++ Enforcer plugin include:

- Sun/Netscape/iPlanet Enforcer plugin
- IIS Enforcer plugin
- Apache 2 Enforcer plugin
- Domino Enforcer plugin
- Oracle Enforcer plugin

With these Enforcer plugins, you must encode the following three forms in `<install_path>/content` directory to use the default character set used by the application rather than UTF-8:

- `Post_accepted.html`
- `Post_password_changed_form.html`
- `Post_redirect.html`

To make this change:

1. Locate the following entry in each of these forms:

```
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
```

2. Change it to:

```
<meta http-equiv="Content-Type" content="text/html;
charset=<site_character_set>">
```

You do not need to make any code changes to these C++-based Enforcer plugins or to the Policy Validator.

⚠ Be aware that once the changes are made, the web server will only be able to handle characters of the specified character set. In order to support all languages, you MUST convert the entire web site to UTF-8.

⚠ The processing of these characters remains a documented issue for the servlet Enforcer plugin. Therefore, ensure that identities are authenticated before the POST data to avoid triggering these POST-over-auth functions.

4 Other Apache Server Integrations

Select Access transparently supports Apache 2.0 web servers with its Apache 2 Enforcer plugin. This chapter describes these integrations.

These other manual integrations include:

- Tomcat 4.1.31 servlet engine with Apache 2 servers over SSL (add-on module deployment)

  ➤ You can also use the servlet Enforcer plugin with Apache/Tomcat built-in deployments. For details, see Chapter 6, Servlet Engine Integrations.

- Apache 2 as a reverse proxy server

Chapter Overview

Table 1 on page 10 lists the web servers Select Access currently integrates with. While the Apache 2.0 web server is listed as a transparently supported configuration, all other Apache deployments require that you read the contents of this chapter. Topics in this chapter include:

- Integration Overview on page 63
- Integrating the Apache 2 Enforcer Plugin with Apache/Tomcat Systems Over SSL on page 68
- Deploying the Apache 2 Enforcer Plugin on page 68
- Deploying Apache Tomcat on Solaris and Linux on page 75

Integration Overview

The integration process of a directory server with Select Access is impacted primarily by the type of web server you are using and whether or not an Enforcer plugin is installed and configured transparently for the server's host computer. Other factors that can affect the deployment of Select Access with your web server include:

- Which authentication mechanism you require
- What level of protection the web servers resources need
- Whether your server performs virtual hosting
- How site data is used with respects to personalization
- Whether you require single sign-on (SSO) across single or multiple domains

➤ The Select Access Apache Enforcer is 64 bit and needs to be run inside a 64 bit Apache 2 server. A 32 bit Apache server can run on HPUX-64 but it will fail to start after the Select Access Apache Enforcer is loaded.

Integration Tasks

The following integration tasks are generic to all web servers, whether or not installation and configuration occurs transparently or otherwise.

Task 1: Prepare your web server for installation.

To make your installation and configuration of your web server and corresponding plugin integrate and operate more effeciently, there are some steps you can consider. For details, see Preparing Your Web Server for Integration on page 39.

Task 2: Install and configure the Apache 2 Enforcer plugin for you deployment of Apache.

Depending on your deployment of Apache, the steps required to install the required Enforcer plugin files will vary. For details, see Installing the Apache 2 Enforcer Plugin on page 65.

➤ If your web server is a host to virtual domains, there are specific configuration tasks you must perform when you configure your plugin. For details, see Securing Virtual Domains on page 44.

Task 3: For Apache 2.0 and Apache/Tomcat with SSL deployments only, integrate the Enforcer plugin with your server.

After you install the corresponding Enforcer plugin for your server, you need to configure the plugin. In some cases, you may need (or want) to manually configure your web server to load the Enforcer plugin. However, in most cases, the Setup Tool manages configuration details automatically. For details, see Deploying the Apache 2 Enforcer Plugin on page 68.

➤ Reverse proxy deployments are integrated transparently via the Setup Tool.

Task 4: Add the server and its resources to the Policy Matrix. Set up authentication and access policy as required.

Determine what you need to do to your web server and its resources to get content protected. The procedure for accomplishing this is not so much contingent upon your server as it is your deployment environment, your security concerns and your business requirements.

For details, see Securing Your Web Server's Resources on page 46.

Task 5: Customize any forms you require for user authentication.

If you plan to perform form-based login, you may want to customize default form templates shipped with Select Access. There are no differences for servers that require manual installation and integration than those that are transparently supported. For details, see Using Forms on Enforcer-protected Web Servers on page 47.

Task 6: For Apache/Tomcat deployments over SSL, test your integration.

Because this integration involves three tiers of components, you want to ensure all plugins, applications, and so on are functioning as expected. Details of testing this integration with the Apache 2 Enforcer plugin is similar to that of the servlet Enforcer plugin used with the Tomcat servlet engine for built-in server deployments without the Apache server. For details, see Testing Your Deployment on page 98.

Installing the Apache 2 Enforcer Plugin

The Enforcer plugin is just one component of a scalable Select Access access management solution for your network. For web servers, the Enforcer plugin is the agent for Select Access on the host computer, and acts as the primary point of integration for these servers.

Unlike the Enforcer plugins for transparently supported web servers, the other Enforcer plugin modules for other web servers are not always installed with the Select Access installer, nor are they necessarily seamlessly integrated with a wizard in the Setup Tool.

For server integrations that are either completely or partially manually integrated, Enforcer plugin deployments typically consist of the tasks described in Installing Plugin Files on page 65.

Installing Plugin Files

You can install the Enforcer plugin required for your server in one of the following ways:

- CD: Some modules are available from the product CDs. By default files for uncommonly-used Enforcer plugins can be found in the `solutions` folder.
- Sales team: When the integration work takes place in between releases of Select Access, you can contact your sales person to obtain a copy of the file you require.

For details unique to your web server, see:

- Transparently Installing and Configuring the Apache 2 Enforcer Plugin for Apache/ Tomcat Systems Over SSL on page 65
- Installing and Configuring the Apache 2 Enforcer Plugin for a Reverse Proxy Server on page 67

Transparently Installing and Configuring the Apache 2 Enforcer Plugin for Apache/Tomcat Systems Over SSL

Select Access with Tomcat 4.1.3.3 over Apache 2 allows you to build a three-tier system, as shown in Figure 6 on page 66. This system:

- Makes the deployment of dynamic JSP pages and servlets easier
- Allows for enhanced transaction management

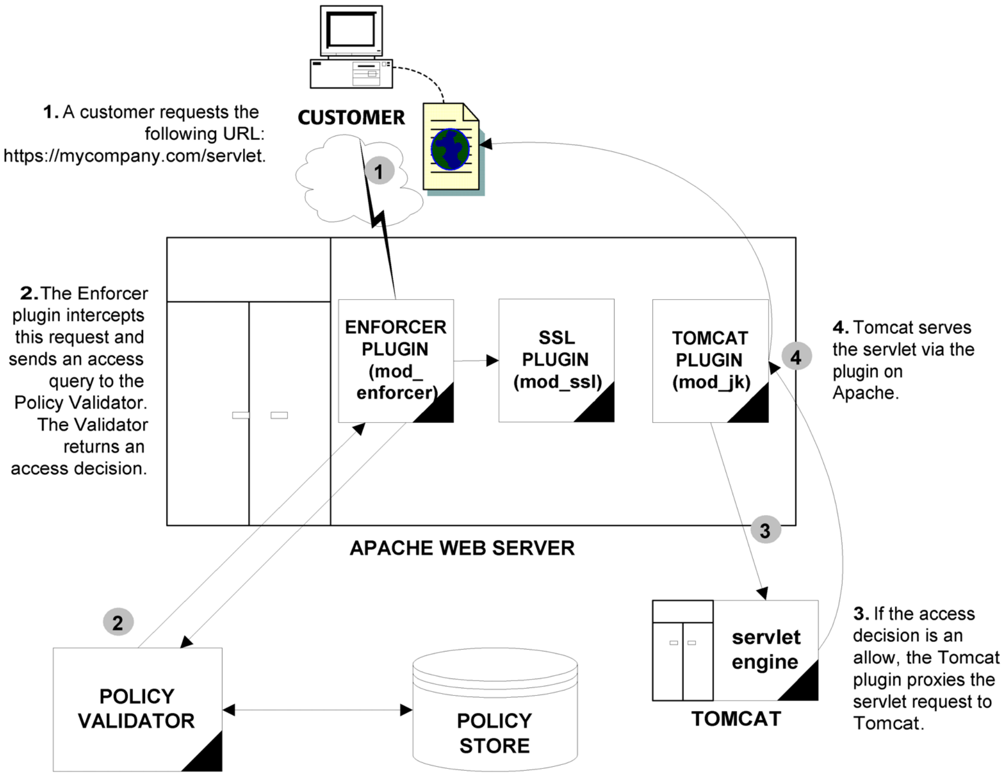

CUSTOMER
1. A customer requests the following URL: https://mycompany.com/servlet.
1
2. The Enforcer plugin intercepts this request and sends an access query to the Policy Validator. The Validator returns an access decision.
ENFORCER PLUGIN (mod_ enforcer)
SSL PLUGIN (mod_ssl)
TOMCAT PLUGIN (mod_jk)
4
4. Tomcat serves the servlet via the plugin on Apache.
APACHE WEB SERVER
3
2
POLICY VALIDATOR
POLICY STORE
servlet engine
TOMCAT
3. If the access decision is an allow, the Tomcat plugin proxies the servlet request to Tomcat.

Figure 6 Integrating Select Access with Apache and Tomcat

This figure shows that you install the Apache 2 Enforcer plugin on the same Apache server as the SSL plugin and the Tomcat proxy plugin. This ensures that all resource requests are handled by the Apache 2 Enforcer plugin, not the Tomcat servlet engine. For details, see Integrating the Apache 2 Enforcer Plugin with Apache/Tomcat Systems Over SSL on page 68.

To install and configure the Apache 2 Enforcer plugin on an Apache web server with a Tomcat plugin

1. Because you are installing the Enforcer plugin on an Apache host, there are specific components you need to preinstall. Ensure you have met the requirements outlined in Preparing Your Web Server for Integration on page 39.
2. Install the Apache 2 Enforcer plugin as described in the *HP OpenView Select Access 6.2 Installation Guide*. Depending on whether your are running the installer from the command line or the user interface equivalent, the process can vary slightly. You can install the Apache 2 Enforcer plugin with or without other components on the same host.
3. When you have installed the Apache 2 Enforcer plugin, the Setup Tool appears. Click **Next** until you reach the setup wizard for the Apache 2 Enforcer plugin. You will need to configure the plugin to minimize the number of extra forked Apache processes that are created when the server receives multiple requests in parallel.
4. On the **General** setup screen, choose the **Custom** configuration option.
5. Configure the Apache 2 Enforcer plugin with the following **Tuning Parameter** values, especially if your Apache web server is using International Components for Unicode (ICU) libraries. For reference purposes, XML parameter name equivalents are included in the brackets.
    - **Wait for Validator to reply no more than** (`ctimeout`) = 60

- **Retry unavailable Validators every** (`rtimeout`) = 30
- **Consider Validator unreachable after** (`KeepAliveTimeout`) = 3-5

6 If you are automatically allowing the Setup Tool to modify Apache's `httpd.conf` file, ensure that the plugin is the last module listed.

7 Manually modify the Apache 2 Enforcer plugin properties in the `httpd.conf` file in the configuration directory. You must make these changes manually because these parameters are not configurable with the Setup Tool.

Installing and Configuring the Apache 2 Enforcer Plugin for a Reverse Proxy Server

The Apache 2.0 Reverse Proxy web server is an HTTP server that has been modified to allow:

- Remote servers to be mapped into the space of the local server
- Apache to adjust the URL in the location header on HTTP redirect responses

As a result, this server does the following:

- Uses caching features to provide load balancing on a heavily-used server.
- Runs outside the firewall to represent a secure content server to outside clients, preventing direct, unmonitored access to your server's data from outside your company. Identities cannot get to the real content server because the firewall passage only allows access to the proxy server.
- Filters client transactions by controlling access to remote servers and protocols and by limiting access to specific documents or sites based on usernames, URLs, and client hostnames (or IP addresses).

To install and configure the Apache 2 Enforcer plugin with an Apache reverse proxy server

Use following steps to build Apache 2 on the Linux and Solaris platforms.

1 Export `APACHE_HOME=/usr/local/apache2.0.55`.

2 Export `JAVA_HOME=/usr/local/java/j2sdk1.4.2_10`.

3 Enter the following code:

- For Linux:

```
./configure --prefix=$APACHE_HOME \
--enable-proxy=shared \
--enable-ssl=shared
make
make install
```

- For Solaris:

```
./configure --prefix=$APACHE_HOME  \
--enable-so \
--enable-proxy=shared \
--with-ssl=/usr/local/ssl \
--enable-ssl=shared
make
make install
```

4 Ensure you have installed Apache with the `mod_proxy` module to the reverse proxy server and added the correct directives for this module. The directives required for this integration are:

> ➤ `mod_proxy` and `mod_jk` are two different ways to connect Apache 2 with Tomcat. If `mod_jk` is used, `mod_proxy` is not needed.

- `ProxyPass`: Allows remote servers to be mapped into the space of the local server.
- `ProxyPassReverse`: Lets Apache adjust the URL in the location header on HTTP redirect responses. This is essential when Apache is used as a reverse proxy to avoid by-passing the reverse proxy because of HTTP redirects on the backend servers which stay behind the reverse proxy.

For detailed instructions, visit `http://httpd.apache.org` and `http://httpd.apache.org/docs/2.0/mod/mod_proxy.html`.

5 Because you are installing the Enforcer plugin on an Apache host, there are specific components you need to preinstall. Ensure you have met the requirements outlined in Preparing Your Web Server for Integration on page 39.

6 Install the Apache 2 Enforcer plugin as described in the *HP OpenView Select Access 6.2 Installation Guide*. Depending on whether your are running the installer from the command line or the user interface equivalent, the process can vary slightly. You can install the Apache 2 Enforcer plugin with or without other components on the same host.

7 When you have installed the Apache 2 Enforcer plugin, the Setup Tool appears. Click **Next** until you reach the setup wizard for the Apache 2 Enforcer plugin. Configure this plugin according to the requirements of your network environment.

8 On the final setup screen, ensure that you check the following two boxes before clicking **Finish** to exit the setup wizard for the Apache 2 Enforcer plugin:

- **Update Web server configuration to load the Enforcer plugin**.
- **Restart Web server.**

Otherwise, your Apache 2 Enforcer plugin is not transparently integrated with the Apache reverse proxy server.

Deploying the Apache 2 Enforcer Plugin

The Enforcer plugin is just one component of a scalable Select Access access management solution for your network. For web servers, the Enforcer plugin is the agent for Select Access on the host computer, and acts as the primary point of integration for these servers.

By installing additional Enforcer plugins for each web server on your network, you can effortlessly coordinate Select Access' deployment. With each new installation, Enforcer plugins are registered in the Select Access directory server. Configuration of all components is therefore centralized and easily managed.

Integrating the Apache 2 Enforcer Plugin with Apache/Tomcat Systems Over SSL

Integrating these three technologies requires that you configure Apache, Tomcat, and Select Access with specific properties and parameters to ensure they function properly as a unit. Specifically, you want to set up your three-tier system so Tomcat cannot accept resource requests indirectly through its built-in web server; as shown in Figure 6 on page 66. Always force resource requests through the Apache server instead. Table 11 lists the integration tasks involved in this deployment of Select Access.

The subsequent sections assume that you have downloaded the binaries for Tomcat v4.1.31 and the source for Apache 2, although you can use more recent versions. If you do use a more recent version, you simply need to rebuild the Apache 2 Enforcer plugin.

Table 11 Integration Tasks for Apache, Tomcat, and Select Access Deployments

Integration Task	Details
1 If you have not already done so, download and install Perl. Without Perl, you cannot build required packages on Apache.	You extract the required `perl-5.8.0.tar.gz` package after you have downloaded it from: `http://cpan.org/src/5.0/`
2 Rebuild Apache to: — Run with Tomcat — Use SSL	To load Tomcat and SSL modules on SSL-enabled Apache servers on page 70
3 Configure the Tomcat servlet engine to minimize the number of connections it can maintain. Tomcat uses an entity called “workers” to run servlets on the Apache web server’s behalf. Worker definitions are added to a `workers.properties` file. This repository of worker definitions is used by Tomcat so it knows where different workers are and which workers it needs to forward requests to.	To create a worker.properties file on page 71
4 Because the Apache 2 Enforcer plugin only intercepts resource requests for the Apache web server, you need to disable Tomcat’s built-in web server and only keep the `ajp13` connector active so it can proxy servlet requests to Tomcat. This action minimizes the number of unwanted direct connections to the Tomcat servlet engine.	To minimize the number of direct connections on page 72
5 Configure Apache to load the Apache 2 Enforcer plugin. You may need to rebuild the plugin if: — You want to avoid receiving harmless warning messages, especially on a Solaris hosts. — You require SSL EAPI support.	To load and rebuild the Apache 2 Enforcer plugin on page 72

To load Tomcat and SSL modules on SSL-enabled Apache servers

1 Download and extract all required packages as listed in Table 12:

Table 12Packages Required

Package	Source	Details
`openssl-0.97a.tar.gz`	`http://www.openssl.org/source/`	The open source SSL library that allows Apache to communicate over SSL.
`jakarta-tomcat-connectors-1.2.15-src.tar.gz`	`http://www.apache.org/dist/tomcat/tomcat-connectors/jk/source/jk-1.2.15/`	The Tomcat plugin. The version of `mod_jk` you require correlates to the version of the Tomcat servlet engine.

2 Build the packages listed in Table 12:

a Build the OpenSSL package by:

— Changing to the directory in which you expanded the OpenSSL libraries. For example, using the default directory, the command for this is:

```
cd openssl-0.9.7a
```

— Parsing the configuration file so that it creates general parameters from it, run the command shown below from this directory.

On Linux:

```
./config -fPic
```

On Solaris

```
./config -shared -s -fPic
```

— Compiling the library, run the `make` command.

— Testing the results of the compiled library, by running the commands `make`, `make test` and `make install`.

b Build the `mod_jk` package by following the procedure unique to your OS:

— To build mod_jk on Solaris on page 71

— To build mod_jk on Linux on page 71

3 Edit the `LoadModule` section of your `httpd.conf` file so that Apache loads the `mod_jk` plugin after `mod_ssl`. The lines below provide an example of these modifications:

```
# load the jk_module
LoadModule    jk_module  libexec/mod_jk.so
AddModule     mod_jk.c

# jk_module configuration properties
JkWorkersFile <workers_properties_file_path>/workers.properties
JkLogFile     /usr/local/apache/logs/mod_jk.log
JkLogLevel    info
JkLogStampFormat "[%a %b %d %H:%M:%S %Y] "
```

```
# directories you want to redirect to Tomcat
JkMount /*.jsp ajp13
JkMount /examples/servlet/* ajp13
```

To build mod_jk on Solaris

1. Change to the Tomcat plugin directory. Using Tomcat's defaults, this directory is:

   ```
   cd jakarta-tomcat-connectors-1.2.15-src/jk/native
   ```

2. Enter the following:

   ```
   APACHE_HOME=/usr/local/apache2.0.55
   export APACHE_HOME
   JAVA_HOME=/usr/local/java/j2sdk1.4.2_10
   export JAVA_HOME
   ./buildconf.sh
   ./configure -with-apxs=$APACHE_HOME/bin/apxs
   make
   make install
   ```

To build mod_jk on Linux

1. Change to the Tomcat plugin directory. Using Tomcat's defaults, this directory is:

   ```
   cd jakarta-tomcat-connectors-1.2.15-src/jk/native
   ```

2. Export the `APACHE_HOME` and `JAVA_HOME` values as environment variables to reflect your specific installation folder for Apache and the JDK. For example, using default installation directories, you would set values with the following command:

   ```
   APACHE_HOME=/usr/local/apache2.0.55
   export JAVA_HOME=/usr/local/java/j2sdk1.4.2_10
   export APACHE_HOME =usr/local/apache
   export JAVA_HOME=usr/local/java
   ```

3. Enter the following command:

   ```
   ./buildconf.sh
   ./configure -with-apxs=$APACHE_HOME/bin/apxs
   make
   make install
   ```

To create a worker.properties file

1. Define the Tomcat install directory. The syntax for this is:

   ```
   workers.tomcat_home=<tomcat_install_path>
   ```

2. Define the Java install directory. This allows Tomcat workers to find the Java files they need to serve Java servlets. The syntax for this definition is:

   ```
   workers.java_home=<jdk_install_path>
   ```

3 Create the worker list. A worker is an instance of a Tomcat process. A worker can use a number of protocols so that it can process servlets that the Apache web server requests. There are many different protocols you can use with Tomcat. However, our worker list only includes the ajp13 worker. This worker is the one that forwards requests to out-of-process Tomcat workers using the ajp13 protocol. For example:

```
worker.list=ajp13
```

4 Define the properties for the ajp13 worker. These properties define the connection information for it. The syntax for a property definition is:

```
worker.<worker_name>.<property_name>=<value>
```

For example, typical properties you would want to define include:

```
worker.ajp13.port=8009
worker.ajp13.host=localhost
worker.ajp13.type=ajp13
```

For more details on other properties available for the ajp13 worker, see Tomcat's *Worker Definition* guide.

To minimize the number of direct connections

1 Deactivate all connectors, with the exception of the ajp13:

a Open the `server.xml` configuration file. By default, you can locate this file in the following directory:

```
<tomcat_install path>/conf/
```

b Disable all HTTP listeners and other unwanted connectors so that Tomcat cannot use its built-in web server. Do this by commenting out all corresponding lines in this file so only the ajp13 connector remains active. XML requires that you use the tags to wrap the lines as shown below:

```
<!-- Disable this line -->
```

The following example shows a non-SSL section that has been disabled with this syntax.

```
<!-- Define a non-SSL HTTP/1.1 Connector on port 8080 -->
   <!--
   <Connector className="org.apache.catalina.connector.http.
   HttpConnector"
         port="8080" minProcessors="5" maxProcessors="75"
         enableLookups="true" redirectPort="4443"
         acceptCount="10" debug="0" connectionTimeout="60000"/>
   -->
```

c Save your changes.

2 Heavily firewall Tomcat's host machine to limit direct `ajp13` requests to those originating from Apache's host computer.

To load and rebuild the Apache 2 Enforcer plugin

1 Make sure the Policy Validator is running.

2 Open your `httpd.conf` Apache configuration file:

```
vi /etc/httpd/conf/httpd.conf
```

3 Add the following lines to the beginning of the `LoadModule` section. These lines ensure that the Apache 2 Enforcer plugin starts after the SSL module has already loaded. Depending on your deployment of the Apache server, this location varies:

- If Apache uses its own access control handler, place the lines above the `enforcer_module` line in the `LoadModule` section.
- Otherwise, add these lines after the `mod_ssl` section. Use either of the following platform-specific lines, as appropriate for your host computer:
  - — For HP–UX:

    ```
    AddModule enforcer_module <install_path>/bin/mod_enforcer.cpp
    ```

    ```
    LoadModule enforcer_module <install_path>/bin/mod_enforcer.sl
    ```
  - — For Linux and Solaris:

    ```
    AddModule enforcer_module <install_path>/bin/mod_enforcer.cpp
    ```

    ```
    LoadModule enforcer_module <install_path>/bin/mod_enforcer.so
    ```

4 If you use SSL, add the following line to export SSL data the Policy Validator requires. Without adding this line, any SSL encryption decision points in your existing rules fail.

```
SSLOptions +ExportCertData +CompatEnvVars +StdEnvVars
```

5 Comment out the `ClearModuleList` line.

6 Save `httpd.conf` and restart the Apache web server. If you are running the Apache web server on HP-UX, you need to start the web server manually or set `LD_PRELOAD` in the environment. For details, refer to Chapter 8, Configuring the Enforcer Plugins, in the *HP OpenView Select Access 6.2 Installation Guide*.

7 If you need to rebuild the Apache 2 Enforcer plugin:

a Change to Select Access' installation directory. Using Select Access' default location, this command is:

```
cd /opt/OV/SelectAccess
```

b Run the `make` command to rebuild the Apache 2 Enforcer plugin.

Securing Your Web Server's Resources

Once you have successfully integrated the Enforcer plugin on your web server's host computer, you can begin securing your resources. The Policy Matrix provides a clear view of all users and resources. It lets you understand the overall policy for specific user and resource combinations, which allows you to answer questions like "Who can access a specific resource?" or "Which resources does an individual have access to?".

Discovering Your Resources

If you have setup your servers as described in Preparing Your Web Server for Integration on page 39, building a tree of resources you want to secure is simplified. To deploy Select Access even more quickly, the Policy Builder includes an innovative scanning mechanism called a Automatic Resource Discovery, that automatically enumerates all network services and resources, such as URLs, dynamic pages and portal links. The results are displayed in the

Policy Matrix and added to the directory in a hierarchical format that best reflects intranet, extranet, or portal resources. For explicit details on this tool, refer to Chapter 4, Organizing Identities and Resources, in the *HP OpenView Select Access 6.2 Policy Builder Guide*.

To secure your resources

1 Add the web server to the Resources Tree in the Policy Matrix, defining the protocol as either HTTP or HTTPS (for SSL). For details, see Chapter 3, Building your Identities and Resources Trees in the *HP OpenView Select Access 6.2 Policy Builder Guide*.

2 Set the access policy for this web server. Pairings of identities, resources and associated policy rules are displayed with policy rule icon: “allow,” “deny” or “conditional”. Policy inheritance allows you to automatically set policy for groups of identities and resources, thereby eliminating the need to manually assign policy for every identity and/or resource.

3 However, with this deployment where you set the policy is as important as what you set the policy to. If you have configured your server to map URLs, set access control on the source URL, not the one mapped to it. For example, if you map all requests from `http://www.abc.com/mycompany/employees/` folder to the `http:/internal123.abc.com/yourcompany/prices/` folder, you need to set permissions on the `/mycompany/employees` folder of the `http://www.abc.com` service. Ensure you set policy on the first resource only. As a rule, the second resource is not managed by Select Access.

> ➤ Good policy setting practice dictates that you always determine identity access with a policy and not an authentication method. When you try to use the authentication service to determine access logic, the final access decision can be unpredictable.

4 Choose the authentication methods used to identify unknown users. Select Access supports multiple authentication types, including user self-registration and passwords, Windows NT, Kerberos, SecurID, tokens and X.509 certificates.

5 Access the web server using HTTP to confirm that the Policy Enforcer plugin is installed and configured correctly. Verify that access is allowed or denied based on the policy you set and the authentication method you have defined.

Once you have set up the web server following these instructions, no one can access resources on this web server unless they first:

- Authenticate themselves as a known user in the directory server
- Have an access policy that gives them access to a protected resource

Deploying Apache Tomcat on Solaris and Linux

Following tools should be installed before doing beginning:

- perl
- automake
- autoconf
- m4
- gcc
- make
- libgcc
- libiconv
- libtool

1 Download the following packages:

Table 13Required Packages

Package	Source
`openssl-0.9.7a.tar`	`http://www.openssl.org/source/`
`jakarta-tomcat-connectors-1.2.15-src.tar`	`http://www.apache.org/dist/tomcat/tomcat-connectors/jk/source/jk-1.2.15/`
`httpd-2.0.55.tar.gz`	`http://archive.apache.org/dist/httpd/`
`jakarta-tomcat-4.1.31.jar`	`http://tomcat.apache.org/download-41.cgi`

2 Enter the following commands:

- For Linux:

```
./config -fPic
make
make test
make install
```

- For Solaris:

```
./config -shared -s -fPic
make
make test
make install
```

3 Build and Install Apache 2:

```
export APACHE_HOME=/usr/local/apache2.0.55
export JAVA_HOME=/usr/local/java/j2sdk1.4.2_10
```

- For Linux:

```
./configure --prefix=$APACHE_HOME \
--enable-proxy=shared \
--enable-ssl=shared
make
make install
```

- For Solaris:

```
./configure --prefix=$APACHE_HOME  \
--enable-so \
--enable-proxy=shared \
--with-ssl=/usr/local/ssl \
--enable-ssl=shared
make
make install
```

Apache 2 uses a shorthand notation for the group ID in `httpd.conf`. If you have problems getting Apache 2 to start on Solaris 8 from a default install, check the Apache error log (`/usr/local/apache2/logs/error_log`) for an error message. If you see one that says something like "`[alert] (22)Invalid argument: setgid: unable to set group id to Group 4294967295`", edit `httpd.conf`, and change the line that says "`Group #-1`" to "`Group nobody`" then start/restart Apache.

4 Build `mod_jk`:

```
export APACHE_HOME=/usr/local/apache2.0.55
export JAVA_HOME=/usr/local/java/j2sdk1.4.2_10

cd jakarta-tomcat-connectors-1.2.15-src/jk/native
./buildconf.sh
./configure -with-apxs=$APACHE_HOME/bin/apxs
make
make install
```

You can now see `mod_jk` under `$APACHE_HOME/module/`.

5 Add following lines in `$APACHE_HOME/conf/httpd.conf`;

```
LoadModule jk_module modules/mod_jk.so
JkWorkersFile conf/workers.properties
JkMount /examples/* ajp13
JkLogFile logs/mod_jk.log
JkLogLevel info
JkLogStampFormat "[%a %b %d %H:%M:%S %Y]"
```

6 Create the workers.properties file with following lines in the $APACHE_HOME/conf directory:

```
worker.list=ajp13
worker.ajp13.port=8009
worker.ajp13.host=localhost
worker.ajp13.type=ajp13
```

7 Disable all HTTP listeners and other unwanted connectors so that Tomcat cannot use its built-in web server. Do this by commenting out all corresponding lines in this file so only the ajp13 connector remains active.

The following example shows a non-SSL section that has been disabled with this syntax.

```
<!-- Define a non-SSL HTTP/1.1 Connector on port 8080 -->
  <!--
  <Connector className="org.apache.catalina.connector.
     http.HttpConnector"
     port="8080" minProcessors="5" maxProcessors="75"
     enableLookups="true" redirectPort="4443"…
```

8 Using mod_proxy instead of mod_jk:

> ➤ mod_jk and mod_proxy are two different ways to connect Tomcat with Apache. If mod_jk is used, mod_proxy is not needed.

a Configure your copy of Apache so that it includes the mod_proxy module.

If you are building from source, the easiest way to do this is to include the --enable-proxy=shared directive on the ./configure command line. Please refer to Step 3.

If mod_proxy is not already added, make sure that you are loading the mod_proxy module at Apache startup time, by using the following directives in your httpd.conf file:

```
LoadModule proxy_module  {path-to-modules}/mod_proxy.so

LoadModule proxy_http_module {path_to_modules}/
mod_proxy_http.so
```

b Include two directives in your httpd.conf file for each web application that you wish to forward to Tomcat 4. For example, to forward an application at context path /myapp:

```
ProxyPass         /myapp  http://localhost:8081/myapp
ProxyPassReverse  /myapp  http://localhost:8081/myapp
```

c Configure your copy of Tomcat 4 to include a special <Connector> element, with appropriate proxy settings, for example:

```
<Connector className="org.apache.catalina.connector.http.
HttpConnector"
         port="8081" ...
         proxyName="www.mycompany.com"
         proxyPort="80"/>
```

9 Enable SSL for Apache 2:

```
mkdir $APACHE_HOME/conf/ssl.key
mkdir $APACHE_HOME/conf/ssl.crt
```

```
openssl req -new -x509 -days 30 -keyout $APACHE_HOME/conf/ssl.key/
server.key -out $APACHE_HOME/conf/ssl.crt/server.crt -subj '/
CN=Test-Only Certificate'
```

10 Start Apache with SSL:

```
cd $APACHE_HOME/bin
./apachectl startssl
```

5 Integrating With .NET Frameworks

To integrate with .NET frameworks, Select Access uses Web Services Enhancements (WSE), the .NET class library for building, integrating, and securing web services. WSE uses web services protocols, including WS-Security, WS-Routing, DIME, and WS-Attachments. WSE integrates with ASP.NET web services, allowing Select Access to provide a simple way to extend the functionality of these protocols.

With support for WS-Security, Select Access reduces time and saves money by distributing and sharing business applications securely across departmental and federated business boundaries. In-depth authorization control of SOAP transactions establishes identity and authorizes access appropriate to relationships to the organization.

Chapter Overview

The architectural model of WSE is based on a pipeline of filters that process inbound and outbound SOAP messages, on both client and service sides.

One trait fo WS-security that Select Access leverages is the reliance on the transport capabilities of HTTP servers to exchange SOAP messages between WSs and their clients.

Topics in this chapter include subjects that describe how to integrate the WSE Enforcer plugin as filters in the IIS web server (on .NET):

- Integration Overview on page 79
- Required Components on page 80
- Integrating the WSE Enforcer Plugin on the Server on page 81
- How the WSE Enforcer Plugin Works on page 86

Integration Overview

Although the Select Access installer and Setup Tool take care of installing and configuring the WSE Enforcer plugin, there are other factors that can affect the deployment of Select Access within a .NET framework:

- Leaving some web services unprotected
- Securing SOAP messages by signing and encrypting them
- Disabling HTTP GETs and POSTs

Integration Tasks

These tasks ensure that you meet several system requirements that are unique to this type of Select Access deployment.

Task 1: Install all the required components.

The Select Access installer checks for specific files. Unless these files exist on the host computer, you cannot install the WSE Enforcer plugin. For details, see Required Components on page 80.

Task 2: Install and configure the WSE Enforcer plugin.

You can install and configure your WSE Enforcer plugin either automatically or manually. When you install and configure this plugin, ensure you understand:

- When and why you need to manually install and configure the WSE Enforcer plugin.
- The importance of configuring the WSE Enforcer plugin to sign and encrypt your WS's SOAP messages.

For details on installing and configuring the WSE Enforcer plugin with the Setup Tool, refer to the *HP OpenView Select Access 6.2 Installation Guide*. For all other configuration details, see Integrating the WSE Enforcer Plugin on the Server on page 81.

Task 3: If you have installed the IIS Enforcer plugin, reconfigure it to ignore SOAP messages.

For IIS servers protected with an IIS Enforcer plugin, make sure you configure IIS Enforcer plugin to ignore SOAP messages. Otherwise, the IIS Enforcer plugin incorrectly attempts to validate the request using the information in the HTTP headers.

To allow SOAP messages to bypass IIS Enforcer plugin security, add the web services' relative URLs to the **Ignored Filenames** list when you configure the IIS Enforcer plugin. For more details on configuring an **Ignored Filenames** list, see Chapter 8, Configuring the Enforcer Plugins, in the *HP OpenView Select Access 6.2 Installation Guide*. For more details on integrating both an IIS Enforcer plugin with the IIS server, see Chapter 3, Transparently Supported Web Server Integrations.

Required Components

During the installation process, the Select Access installer searches the host computer to determine whether the prerequisite files are present. If it does not find them, the installer disables the WSE Enforcer plugin component option in the list of available components. Therefore, you need to have the required WS components installed before installing this plugin. You can, however, install other Select Access components first, if you choose.

Preparing Your Server for WS Deployment

There are two things you need to consider:

- Installing the correct components so that the correct files are detectable by Select Access' installer. For details, see To install components required by the WS framework, below.

- Understanding how credentials are passed in SOAP messages using WSE. For more information on sending security credentials in SOAP message, see your WS's documentation.

To install components required by the WS framework

1. Install the following:
   - The Microsoft .NET Framework SDK 1.1
   - The Web Services Enhancements 2.0 add-on
2. Test your environment to see:
   - If you are able to access your WS with a browser
   - If you have already enabled WSE and SOAP extensions for your WS, try to trigger methods on it with a SOAP client

   For details on either of these steps, refer to your .NET documentation.

Integrating the WSE Enforcer Plugin on the Server

In most cases, the Select Access installer and Setup Tool respectively install and integrate the WSE Enforcer plugin automatically. However, if the installer fails to locate a file it needs, or if you want more control over the integration process, you need to complete these tasks. For details see:

- Manually Installing the WSE Enforcer Plugin on page 83
- Manually Integrating the WSE Enforcer Plugin with the Server on page 83

However, before you begin, you need to understand the importance of signing and encrypting your WS's SOAP messages. Based on the contents of the next section, you can make an informed decision when it comes time to configure your WSE Enforcer plugin.

Configuring the WSE Enforcer Plugin to Sign and Encrypt WS SOAP Messages

You can secure SOAP messages sent by your WS by configuring the WSE Enforcer plugin to sign and encrypt all SOAP messages. For specific details, see the corresponding section that follows:

- To use the WSE Enforcer plugin to sign SOAP messages on page 81
- To use the WSE Enforcer plugin to encrypt SOAP messages on page 82

To use the WSE Enforcer plugin to sign SOAP messages

1. Choose your signing method:

   Certificate: The WSE Enforcer plugin can use a certificate and its private key to sign outgoing messages. Before you configure your WSE Enforcer plugin use a certificate for signing, make sure you:

   a. Install the certificate.

   b. Ensure you set the correct permissions on the folder containing the private key or on the private key file itself. By default, private keys are saved in the following folder:

```
C:\Documents and Settings\All Users\Application
Data\Microsoft\Crypto\RSA\MachineKeys
```

Password: The WSE Enforcer plugin can use a password to sign outgoing messages. It creates an element called a Username Token from the values you configure.

2 Run the setup wizard for the WSE Enforcer plugin from the Setup Tool.

3 On the **Sign SOAP XML** setup screen, configure the WSE Enforcer plugin to use the signing method you have chosen.

> For information on setting SOAP signing options, see Setting up SOAP Message Signing in the *HP OpenView Select Access 6.2 Installation Guide*.

- If you are using certificates: You need to configure the:
  - **Certificate Store**: The location in which WSE Enforcer plugin looks for X.509 certificates when it attempts to retrieve or verify a certificate (either Local Machine or Current User).
  - **Certificate Subject**: The substring of the subject of the certificate which can be used to uniquely identify the certificate.
- If you are using password: You need to configure the values that make up the Username Token:
  - **Username** and **Password**
  - **Password Option**: The format of the password: plain text, hashed, or not at all.

    Select the plain text format, if you are using a password to sign SOAP messages sent to Select Access-protected WSs. This is because the Select Access password provider simply returns the password string found in the Username Token. Therefore, all Username Tokens sent to the protected web service must contain the password in clear text.

    Select other formats (hashed or not at all, if you have already share your password with receivers of the message. Otherwise they cannot validate the Username Token.

    > For more about Username Tokens and password options, certificate troubleshooting, see the WSE documentation.

4 Determine whether or not you also need the WSE Enforcer plugin to also validate incoming SOAP messages. Checking the **Require all incoming messages to be signed** box, allows the plugin to see if the SOAP Body of the request message was signed.

5 If you are using plain text passwords to sign messages, make sure you run the Setup Tool and (re)configure Select Access to communicate over SSL. For details, see Chapter 5, Configuring the Administration Server, in the *HP OpenView Select Access 6.2 Installation Guide*.

To use the WSE Enforcer plugin to encrypt SOAP messages

For added security, you can encrypt outgoing SOAP messages. Outgoing messages are encrypted using the same certificates used to sign incoming SOAP requests. Therefore, encryption relies on the incoming message to be signed by a valid certificate which supports digital signatures.

For information on setting the SOAP encryption option, see Setting up SOAP Message Encrypting in the *HP OpenView Select Access 6.2 Installation Guide*.

Manually Installing the WSE Enforcer Plugin

When you install the WSE Enforcer plugin, the Select Access installer searches for the GAC file (`gacutil.exe`). Once it knows the location of this executable file, it automatically installs the necessary assemblies in the appropriate location.

However, if the installer is unable to locate the file, you must manually add the assemblies to the GAC.

In order for the WSE Enforcer plugin to function, two assemblies are needed by the .NET Framework Global Assembly Cache (GAC):

- `InteropENFORCERLib.dll`
- `WSEEnforcer.dll`

To manually add the assemblies to the GAC

1 Click **Start** → **Programs** → **Administrative Tools** → **Microsoft .NET Framework Configuration**. The **.NET Framework Configuration** window opens.

2 In the left pane, click the **Assembly Cache** entry.

3 Add the WSE Enforcer plugin assemblies to the GAC. To add the assemblies:

  a Select **Action** → **Add**. The **Add an Assembly** dialog box appears.

  b Select the `Interop.ENFORCERLib.dll` and click **Open**.

  c Select **Action** → **Add**. The **Add an Assembly** dialog box appears.

  d Select the `WSEEnforcer.dll` and click **Open**.

4 Close the **.Net Framework Configuration** window.

Manually Integrating the WSE Enforcer Plugin with the Server

When you use the Setup Tool to configure the WSE Enforcer plugin and check **Update Configuration Files** on the **Update Configuration** setup screen, the Setup Tool automatically updates the `web.config` file. Unlike Component Configuration in the Policy Builder, however, the Setup Tool not only defines the configuration for the WSE Enforcer plugin, but also modifies the web server/Application server's `web.config` file. Therefore, you must integrate the WSE Enforcer plugin with your server manually if:

- You use the Component Configuration utility in the Policy Builder.
- The Setup Tool fails to modify the necessary configuration files.
- You have already deleted the password provider defined in the configuration file, which prevents the Setup Tool from writing to the file.
- You installed the WSE Enforcer plugin in Console mode and cannot use the Setup Tool to integrate the plugin.
- You want more control over the configuration file modification process and would prefer to make the changes yourself.

➤ For more information on using these utilities to configure the plugin, see Chapter 8, Configuring the Enforcer Plugins, in the *HP OpenView Select Access 6.2 Installation Guide* and Chapter 16, Modifying Components' Central Configuration Parameters, in the *HP OpenView Select Access 6.2 Policy Builder Guide*.

To manually modify the web.config file

1. Add an entry for the WSE Enforcer plugin so it is added to sequence of SOAP filters for your server. The syntax of this entry is:

2. Add an entry for the Enforcer plugin's input and output filters. For example:

```
<microsoft.web.services>
   <filters>
       <input>
          <add type="com.hp.ov.selectaccess.enforcer.wse.InputFilter,
            WSEEnforcer, PublicKeyToken=...., Culture=neutral,
            Version=1.0.0.0" />
       </input>

       <output>
           <add
type="com.hp.ov.selectaccess.enforcer.wse.OutputFilter,
           WSEEnforcer,  PublicKeyToken=..., Culture=neutral,
           Version=1.0.0.0" />
      </output>

   </filters>
Make Select Access the password provider for your server. For example:
   <security>
       <passwordProvider
       type="com.hp.ov.selectaccess.enforcer.wse.PasswordProvider,
       WSEEnforcer, PublicKeyToken=..., Culture=neutral,
Version=1.0.0.0" />
   </security>
```

3. If you configured the WSE Enforcer plugin to sign data, update the `storeLocation` attribute of the `<X509>` element with the location of your signing certificate. The WSE Enforcer plugin also uses the same certificate location as part of its configuration details that are saved to the directory server.

   If you either change the location or name of the certificate used for data signing, ensure you either use the Setup Tool to update the configuration details. Only the Setup Tool can write changes to the `web.config` file. Alternatively, you can also modify this file by hand.

   For example, you would add the following line below the entry for the password provider for you server:

```
   <x509 storeLocation="<myHost>:<path_to_cert>" />
```

4. To ensure the WSE Enforcer plugin filters are only triggered by HTTP POSTs that contain SOAP requests, disable requests with:

   - HTTP GET
   - HTTP POSTs without SOAP

   You can disable these methods this by adding the following entry:

```
   <webservices>
       <protocols>
```

```
         <remove name="HttpPost" />
      <remove name="HttpGet" />
    </protocols>
</webservices>
```

Rather than disabling methods for each WS, Microsoft recommends that you disable HTTP-GET and HTTP-POST methods on the *entire* machine and only enable HTTP-POST via SOAP to secureWS. For more information, refer to Microsoft's SOAP documentation.

To manually modify the machine.config file

1 Open the `<`*`.NET_install_path`*`>\Framework\<`*`version`*`>\CONFIG\machine.config` file.

2 To configure the identity under which ASP. NET runs on your machine, define the `<processModel>` element. The values vary depending on the platform of your IIS server:

- For Windows 2000 servers: ASP .NET runs under the `ASPNET local` identity account.
- For Windows 2003 servers: When ASP. NET runs under `Network Service` identity account. This is because IIS 6 is a worker process in isolation mode (which is the default setting). The IIS 6 process model is used and the settings of the `<processModel>` element in the `machine.config` file are ignored.

  To learn how to map `machine.config` settings to the IIS 6 application pool settings when running in worker process isolation mode view, refer to Microsoft's documentation.

This allows the WSE Enforcer plugin to give the appropriate accounts access to the private keys.

To Unprotect Web Services

To remove the WSE input and output filter from the `web.config` configuration files of the affected services.

1 In the **Select Web Services** dialog, deselect the WSs that you no longer need to secure.

How the WSE Enforcer Plugin Works

Once you have integrated the WSE Enforcer plugin on your server, it should receive incoming SOAP requests. The request triggers the following actions in the plugin.

1 The WSE Enforcer plugin constructs a Policy Validator query from the request data:

- The WS's URL from the original request.
- The resource name from the SOAP body. If the SOAP body contains more than one child, then the first child will be assumed to be the target resource.

2 The WSE Input filter then extracts any credentials (the Username Token, or Binary X509 Security Certificate Token) included the SOAP request. If more than one set of credentials are included in the SOAP request, the filter uses the first one and ignores all other.

3 Extracted credentials are added to the query.

4 The Policy Validator extracts the credentials in the query and forwards them to any of the supported authentication methods for the .NET framework. Table 14 lists these methods and which extracted credential type are used.

Table 14 Authentication Methods for the .NET Framework

Supported Method	Token Used for Credentials
Password	Username Token
Certificate	X509 Binary Security Certificate Token
SecurID	Username Token only[a]
Windows Kerberos	Username Token + the domain name prepended to the Username
Windows NTLM	Username Token + the domain name prepended to the Username

[a] New PIN cannot be supported with this token as there is no way to get a new PIN.

➤ For subsequent incoming SOAP requests, the WSE Enforcer plugin extracts all cookies from the HTTP and SOAP headers and sends them to the Policy Validator for authentication.

5 Depending on whether authentication passes or fails, the actions the WSE Enforcer plugin takes after that decision varies. For details, see either:

- When the Policy Validator Allows the Request on page 86
- When the Policy Validator Denies the Request on page 87

When the Policy Validator Allows the Request

If the Policy Validator returns an `ALLOW`, the WSE Enforcer plugin performs the following actions:

1 It extracts all cookies from the reply.

2 The cookies the WSE Enforcer plugin extracts are sent to the WS client:

- All Policy Validator cookies are added to the HTTP response header.
- Only the `PolicyUser` cookie is added to the SOAP header and returned in the SOAP response message.

Table 15 shows what the format of the SOAP XML could be.

Table 15Sample SOAP Header

```
<soap:Header>
<sawss:PolicyUser xmlns:sawss="http://www.hp.com/SelectAccess/
webservices/security">AgAAAAAAPzF5......</sawss:PolicyUser>
.....
</soap:Header>
```

When the Policy Validator Denies the Request

If the Policy Validator fails authentication, either because the client did not send any credentials in the SOAP message, or the credentials it did send are invalid, the WSE Enforcer plugin takes the subsequent actions:

1 Depending on whether or not the credentials were sent:

- If the credentials were not sent: The plugin informs the client to send the credentials in the next request. The WSE Enforcer plugin uses a policy assertions contained within a policy statement to express the authentication requirements of the web service. WS-Policy specifies the XML grammar of the policy statement.
- If invalid credentials were sent: The plugin determines the type of authentication service needed for successful authentication in the reply and uses that to collect new credentials; otherwise access is denied.

Table 16 shows a sample deny response, with additional authentication information supplied.

Table 16Sample Policy Validator Deny Response

```
<PolicyValidatorReply>
.
.
.
  <PROPERTY NAME="action">DENY</PROPERTY>
  <PROPERTYLIST NAME="authentication_server_types">
     <PROPERTY NAME="authentication_method">securID</PROPERTY>
     <PROPERTY NAME="authentication_method">password</PROPERTY>
  </PROPERTYLIST>
.
.
.
</PolicyValidatorReply>
```

2 For every authentication method returned in the Policy Validator reply, the WSE Enforcer plugin adds new `<AuthType>` element as part of the Select Access namespace under `<Policy>`. The `<Policy>` XML that the WSE Enforcer plugin creates from the example Policy Validator reply is shown in Table 17.

Table 17Sample <Policy> XML

```
<soap:Header>
<wsp:Policy xmlns:wsp="http://schemas.xmlsoap.org/ws/2002/12/policy">
<wsse:SecurityToken TokenType="wsse:UsernameToken"
wsp:Usage="wsp:Required" xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/
12/secext" />

<sawss:AuthType xmlns:sawss="http://www.hp.com/SelectAccess/
webservices/security">securID</sawss:AuthType>

</wsp:Policy>
...
</soap:Header>
```

3 The WSE Enforcer plugin subsequently returns a SOAP fault in the SOAP response message. The WS client, on recognizing the fault, should look up the Policy statement and attempt to send the correct credentials in the next SOAP request.

Table 18 lists the fault codes the WSE Enforcer plugin can return.

Table 18 Fault Codes Returned

Name	Description
`Client.FailedAuthentication`	The client did not sent valid credentials and could not be authenticated.
`Server.UnableToAuthenticate`	The WSE Enforcer plugin is unable to contact the Validator and/or unable to authenticate the client.
`Client.UnsignedSOAPRequest`	The incoming SOAP request was not signed and all the web service requests only handles signed requests.
`Client.EncryptionTokenNotFound`	The received SOAP request is signed but the signing token cannot be used to encrypt the outgoing SOAP response.
`Client.AccessDenied`	The client is denied access (perhaps because of a suspected Evil URL).
`Server.CannotEncryptSOAPResponse`	The WSE Enforcer plugin was unable to encrypt the SOAP response body using the token used to sign the incoming SOAP request.
`Server.CannotSignSOAPResponse`	The signing certificate could not be obtained from the certificate store or could not be used to sign the outgoing SOAP response message.

6 Servlet Engine Integrations

Select Access' servlet Enforcer plugin provides Java Servlet technology a simple, consistent mechanism for extending the security of a servlet engine. Whether the servlet engine is built into an application server, or whether the server requires an add-on module, the servlet Enforcer plugin secures these engines by intercepting identity requests irrespective of how the servlet engine is started.

Chapter Overview

The architectural model of servlets is based on a pipeline of filters that process inbound and outbound HTTP requests. The servlet Enforcer plugin is built on this model, so that it can easily intercept and transform requests, as well as modify a response. Supported servers include:

- The Apache/Tomcat application server (built-in deployment)

  ➤ You can also use the Apache 2 Enforcer plugin with the Apache web server and the Tomcat Servlet Engine in an add-on module deployment with SSL enabled, rather than the servlet Enforcer plugin. For details, see Chapter 4, Other Apache Server Integrations.

- The Apache Axis SOAP engine with Tomcat
- The BEA WebLogic application server
- The IBM WebSphere application server

Topics in this chapter include subjects that describe how to integrate the servlet Enforcer plugin as filters on any of the supported servers that use a servlet engine irrespective of how it was implemented (that is, built-in vs. module add-on):

- Integration Overview on page 90
- Transferring Servlet Enforcer Plugin Files on page 91
- Integrating the Servlet Enforcer Plugin on page 92
- Additional Integration Issues: Tomcat with Apache Axis on page 95
- Configuring Select Access to Protect Servlets on page 96
- Deploying the Sample Web Application with the Servlet Filter on page 97
- Testing Your Deployment on page 98

Integration Overview

Servlets are programs that respond to client requests within a server-side application. Servlets are not restricted to the HTTP protocol; however, it is the protocol they are most commonly associated with. Therefore, servlet engines are often modules used as part of an application server environment.

Although the Select Access installer and Setup Tool take care of installing and configuring the servlet Enforcer plugin, there are other factors that can affect the deployment of Select Access within a servlet engine framework:

- Whether deployment descriptors or startup scripts are used to integrate the plugin's filters on the servlet engine.
- You use JRE 1.4.

Integration Tasks

With the exception Task 3, the following integration tasks are generic to all supported servlet engines. These tasks ensure that you meet several system requirements that are unique to this type of Select Access deployment.

Task 1: Install the files that make up the servlet Enforcer plugin.

This integration requires that you use all the classes used to create a servlet Enforcer plugin. Depending on what server technology you are integrating with, the integration of this plugin also makes use of Sun Java JAR files. For details, see Transferring Servlet Enforcer Plugin Files on page 91.

Task 2: Integrate the servlet Enforcer plugin.

All servlet engines require that you integrate the filters by modifying a file by hand to include Select Access-specific details to it. Two methods are supported depending on your server. For details, see Integrating the Servlet Enforcer Plugin on page 92.

➤ If you are hosting the Apache Axis SOAP engine with Tomcat, there are additional issues you must take note of. For details, see Additional Integration Issues: Tomcat with Apache Axis on page 95.

Task 3: Configure Select Access to protect applications.

Because servlet deployments are unique when compared to other more traditional integrations, you do not need to configure other components in addition to the servlet Enforcer plugin. This ensures that the servlet engine and its resources are properly protected. For details, see Configuring Select Access to Protect Servlets on page 96.

Transferring Servlet Enforcer Plugin Files

The Select Access installer does not transparently install the servlet Enforcer plugin on the host machine. However, it does install the required filters to default directories if you choose to install the servlet Enforcer plugin. You must then copy these files to the appropriate location depending on your servlet engine.

Files Needed to Enforcer-Protect Servlets

Table 19 lists the files you need to manually deploy on the host machine. The installation location varies depending on the servlet engine you are using. For details on where to deploy these files, see To manually install the servlet Enforcer plugin on page 91.

Table 19 Servlet Enforcer Files

Filename	Where to Get it

`EnforcerAPI.jar`
`jakarta-oro-2_0.jar`
`mail.jar`
`servletenforcer.jar`
`commons-pool-1.2.jar`
`selectauditclient.jar`
	`<SA_install_path>/shared`

`shared.jar`
`xml.jar`
`castor-0.9.3.19-xml.jar`
`msgsresources.jar`
`bcprov-jdk14.jar`
`protomatter.jar`
`jdom.jar`
`ldapjdk.jar`
`xercesImpl.jar`
`xml-apis.jar`
	`<SA_install_path>/shared/jetty/policy_builder/protected`

To manually install the servlet Enforcer plugin

1. Install the servlet Enforcer plugin. For installation details, refer to the *HP OpenView Select Access 6.2 Installation Guide*.
2. Locate the files as described in Table 19 on page 91, and copy them to the following directories, depending on your servlet engine type:
   - For the Tomcat servlet engine, copy all files to the `<Tomcat_install_path>/common/lib` folder.
   - For the WebLogic servlet engine, create a folder named `sa` in the `<WebLogic_install_path>/server/lib` folder and copy all files to this location.
   - For the WebSphere servlet engine, copy all files to the `<Websphere_install_path>/lib/ext` folder.

Integrating the Servlet Enforcer Plugin

For the servlet Enforcer plugin to be completely integrated into your system, it has to be implemented as a series of filters on the host in question. Depending on the servlet engine you are using, the method of integration varies:

- For WebLogic: Integrating with deployment descriptors. This allows you selectively target which web applications you'd like to protect with Select Access. Unlike other Enforcer deployment methods, you can choose which resources need to be secured, rather than globally securing all resources by default. For details, see Modifying Deployment Descriptor Files on page 92.
- For Websphere: Integrating with startup scripts. This allows you to load the servlet's filters when the engine starts. By default, all resources are secured.

Modifying Deployment Descriptor Files

For each web application that you want to restrict identity access to, you need to modify its `web.xml` to use an instance of a servlet Enforcer plugin. If there are multiple applications that require Enforcer protection, the engine loads the servlet Enforcer plugin once at startup, and then creates a new instance of it in memory for each web application.

Use the `web.xml` file to define:

- Where the application can find the servlet Enforcer plugin filters.
- Map the servlet Enforcer plugin to a URL pattern, so the servlet Enforcer plugin knows which URL patterns it needs to protect. Without this information, the application and its resources cannot be Enforcer-protected.

➤ Tomcat's administration tool does not allow you to change this file. You must make this change manually.

To modify the web.xml file

1. Create a new `<filter>` section, and define the servlet Enforcer plugin as one of your web application's filters. Do this by adding the following sections:
   - Define the filter name and filter class of the servlet Enforcer plugin.
   - Add the initialization parameter below this `<filter>` definition.

   If the application requires filters other than the servlet Enforcer plugin, ensure you load the servlet Enforcer plugin before all others.

   ➤ Do not add a path to the name you configure for the servlet Enforcer plugin's configuration file. Select Access' global configuration file contains a parameter called `SELECTACCESS_CONFIGS`. This parameter defines the path for all Select Access configuration files. When the servlet Enforcer plugin is loaded, it gets the path information from this file.

2. Create a new `<filter-mapping>` section below the `<filter>` section and define a new mapping for the servlet Enforcer plugin filter you just created. Do this by defining the following sections:
   - Define the filter map's name.

- Define the URL pattern that the servlet Enforcer plugin filter must be mapped to. HP recommends that you configure a pattern of "`/*`". This pattern ensures that the servlet Enforcer plugin is invoked for every requested resource associated with that specific web application.

Sample web.xml file with Select Access-specific changes

When you have completed the steps listed in the previous section, your `web.xml` will be similar to the one shown below:

```
<?xml version = "1.0" encoding = "ISO-8859-1"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application
 2.3//EN" "http://java.sun.com/j2ee/dtds/web-app_2_3.dtd">

<web-app>
<!-- Define the filters within the Web Application -->
  <filter>
    <filter-name>servletFilter</filter-name>
    <filter-class>com.hp.ov.selectaccess.enforcer.servlet.ServletFilter
    </filter-class>

    <init-param>
      <param-name>enforcer_conf</param-name>
      <param-value>enforcer_servlet.xml</param-value>
    </init-param>

  </filter>

<!-- Map the filter to a Servlet or URL -->
  <filter-mapping>
    <filter-name>servletFilter</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>

<!-- Define the Servlets within the Web Application -->
  <servlet>
    <servlet-name>
    User HTTPsession Servlet
    </servlet-name>
    <servlet-class>
    com.mycompany.servlets.HTTPsession
    </servlet-class>
  </servlet>


<!-- Define Servlet mappings to urls -->

  <servlet-mapping>
    <servlet-name>
    User HTTPsession Servlet
```

```
     </servlet-name>
    <url-pattern>
    /Usersession
    </url-pattern>
  </servlet-mapping>

</web-app>
```

Modifying Startup Scripts for WebLogic Deployments

WebLogic requires that you define the classpath where the filter's files are located. It implements this requirement by through a startup script that you need to modify so that it loads the servlet Enforcer plugin installed on your host machine at runtime.

To modify the startWebLogic.cmd file

1 Open your startup script for the WebLogic. By default the name of startup script is `startWebLogic.cmd`, which is stored in the domain directory you configured with the WebLogic Configuration wizard.

2 Create an `SA_LIB` variable with a path value to the Select Access folder you created in Files Needed to Enforcer-Protect Servlets on page 91. Using the example provided in that step, the variable added to the startup script is:

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

3 Create an `SA_CLASSPATH` variable with a value to the classpath for all Select Access Jar files listed in Table 19 on page 91. Without these paths, the WebLogic server would not know where to find the files for the Enforcer plugin. For example:

```
set
SA_CLASSPATH=%SA_LIB%;%SA_LIB%\jce1_2_2.jar;%SA_LIB%\local_policy.jar
;
%SA_LIB%\US_export_policy.jar;%SA_LIB%\jakarta-oro-2_0.jar;%SA_LIB%\
mail.jar;
%SA_LIB%\castor-0.9.3.19-xml.jar;%SA_LIB%\EnforcerAPI.jar;%SA_LIB%\
jdom.jar;
%SA_LIB%\SAPrinciple.jar;%SA_LIB%\commons-pool-1.2.jar;
%SA_LIB%\selectauditclient.jar;%SA_LIB%\msgsresources.jar;
%SA_LIB%\bcprov-jdk14.jar;%SA_LIB%\xml-apis.jar;%SA_LIB%\ldapjdk.jar;
%SA_LIB%\protomatter.jar;%SA_LIB%\shared.jar;%SA_LIB%\xercesImpl.jar;
%SA_LIB%\xml.jar;
```

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

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

5 Save the changes to this file.

Additional Integration Issues: Tomcat with Apache Axis

Because Apache Axis is essentially a SOAP engine for constructing SOAP processors such as clients, servers, gateways, and so on, these resources need to be Enforcer-protected with its own Axis Enforcer plugin. The Apache Axis server is typically used in a .NET framework for hosting web services. However, a single server can jointly serve servlets and web services from a single host computer.

Preventing Redundancy and Ensuring Performance

Other Select Access deployments do not require two disparate Enforcer plugins for a single-server environment. However, this unique deployment on Tomcat takes advantage of both the servlet Enforcer plugin and Axis Enforcer plugin. This dual installation requires additional integration steps you must undertake to ensure that both function properly by ensuring there is no overlap in the functions they perform.

To integrate the Axis Enforcer plugin

1 Install the Axis Enforcer plugin on the Tomcat machine.

2 From the Setup Tool:

a Run the configuration wizard for the Axis Enforcer plugin and configure a list of ignored filenames for it. This ensures that the Axis Enforcer plugin does not protect the same resources protected by the servlet Enforcer plugin. For configuration details, see To create a list of ignored filenames in the *HP OpenView Select Access 6.2 Installation Guide*.

b Run the configuration wizard for **Custom Settings** and configure a variable for it:

```
EXPORT_POST_DATA
```

This ensure you get correct POST behaviors with Select Access authentication forms. For configuration details, see Chapter 9, Configuring Custom Settings, in the *HP OpenView Select Access 6.2 Policy Builder Guide*.

3 To prevent Tomcat's class loaders from loading the same Enforcer files twice (once when loading the jars deployed with the 'servletfilter and again when loading the Jars deployed with the Axis Enforcer plugin), delete the following Jars from the axis/web-inf/lib directory:

```
jakarta-oro-2_0.jar
ldapjdk.jar
msgresources.jar
bcprov-jdk14.jar
xml.jar
EnforcerAPI.jar
jdom.jar
protomatter.jar
shared.jar
xercesImpl.jar
xml-apis.jar
mail.jar
```

4 In order for virtual servers to function properly and reliably, add an additional parameter to your web.xml file that defines virtual hosting of the Axis engine:

```
<init-param>
<param-name>virtual_hostname</param-name>
<param-value><Axis_service></param-value>
</init-param>
```

The host name you use here will be used as the service name sent to the Policy Validator.

Configuring Select Access to Protect Servlets

Once you have the servlet Enforcer plugin integrated into your servlet environment, you need to configure all Select Access components to function correctly with this technology. Table 20 lists the steps required to configure Select Access to function with servlet applications and protect them correctly.

Table 20 Configuring Select Access for Servlet Environments

Task	Important Details
1 With the Setup Tool create a template XML file by running the Generic Enforcer plugin wizard. Use this tool to set up configuration parameters required by the servlet Enforcer plugin.	• Run the Setup Tool and launch the Generic Enforcer plugin wizard. • Use the wizard to output parameters to a bootstrap configuration file and save it to the server's root folder: `<SA_install_path>/bin/enforcer_servlet.xml` For details, refer to the *HP OpenView Select Access 6.2 Installation Guide*.
2 Add the servlet engine as a service in the Policy Matrix.	You can add the servlet engine either manually or use the discovery tool. However, since you are currently adding one service, HP recommends you add it manually. For details, refer to the *HP OpenView Select Access 6.2 Policy Builder Guide*.
3 Add all Enforcer-protected servlet applications (and all of their required HTML pages, JSP scripts, graphics, and so on, that are part of this application) as resources below the service you added in step 2.	You can add servlets and its resources either manually or use the discovery tool. For details, refer to the *HP OpenView Select Access 6.2 Policy Builder Guide*.
4 Configure your authentication and authorization (access) policies for your servlet applications.	• Enable SelectAuth for the service entry you created in step 2; otherwise, Select Access cannot protect resources for that service. • Assign access policies to your identities and resource combinations. Select Access can handle all authorization requirements you have, when you use any combination of allow/deny/conditional policies.
5 Customize your Select Access authentication forms and copy them to the content directory you configured for your servlet Enforcer plugin. The plugin displays these forms when a resource requires that the identity be authenticated.	• Ensure that the production location you configured for the content directory for your servlet Enforcer plugin contains *all* forms and content files required to Select Access-authenticate your identities. • If your servlet engine uses deployment descriptors, ensure that the directory location you configured for your servlet Enforcer plugin matches the location in all `web.xml` files.

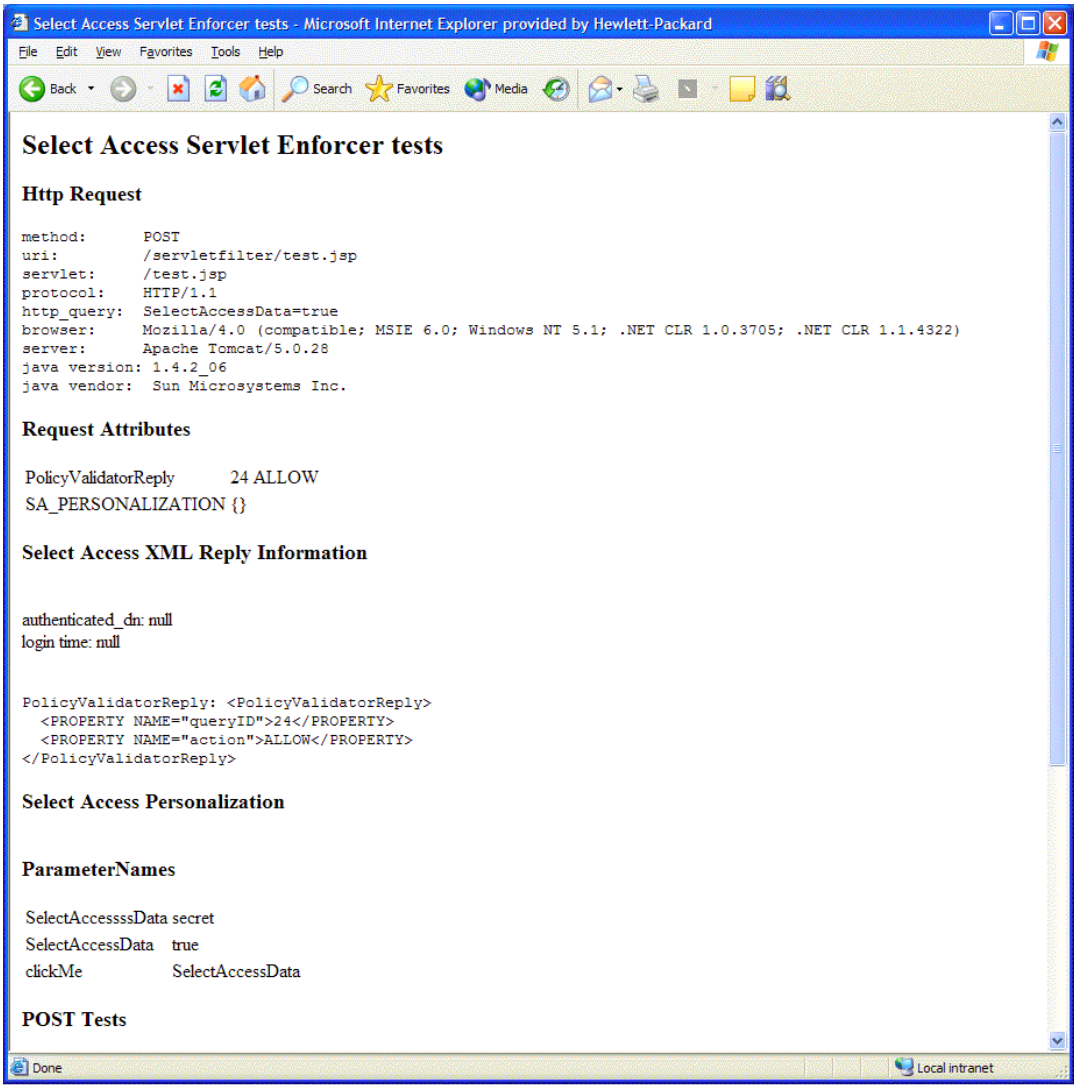
Deploying the Sample Web Application with the Servlet Filter

`servletfilter.war` is a pre-packaged web application that demonstrates the `ServletFilter`. This web application file is not installed with other components; however, HP has included the file on the product CDs. To use the `ServletFilter`, you can deploy the filter on a web server or application server.

To deploy the application

1. Find the `servletfilter.war` file. The file is located on the product CDs in the `/servlet` folder.
2. Deploy the sample application to application server. Follow the vendor-specific instructions for deploying filters for your server.
3. Start the Administration server and run the Setup Tool to configure the servlet Enforcer plugin. Ensure you:
    a. Define the enforcer configuration file name as `enforcer_servlet.xml`.
    b. Choose a **Custom** configuration in order to set a unique enforcer ID for this Enforcer.

   For details, see Configuring the Enforcer Plugins in the *HP OpenView Select Access 6.2 Installation Guide*.
4. Run the Policy Builder to define servlet resources and access policy. For details see the *HP OpenView Select Access 6.2 Policy Builder Guide*.
5. If the access to the sample web application is enabled, you will see a page displaying the request information and Select Access data, as shown in Figure 7.


Select Access Servlet Enforcer tests - Microsoft Internet Explorer provided by Hewlett-Packard
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Done
Local intranet

Select Access Servlet Enforcer tests

Http Request

```
method:      POST
uri:         /servletfilter/test.jsp
servlet:     /test.jsp
protocol:    HTTP/1.1
http_query:  SelectAccessData=true
browser:     Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 1.0.3705; .NET CLR 1.1.4322)
server:      Apache Tomcat/5.0.28
java version: 1.4.2_06
java vendor:  Sun Microsystems Inc.
```

Request Attributes

PolicyValidatorReply	24 ALLOW
SA_PERSONALIZATION {}	

Select Access XML Reply Information

authenticated_dn: null
login time: null

```
PolicyValidatorReply: <PolicyValidatorReply>
  <PROPERTY NAME="queryID">24</PROPERTY>
  <PROPERTY NAME="action">ALLOW</PROPERTY>
</PolicyValidatorReply>
```

Select Access Personalization

ParameterNames

SelectAccesssData	secret
SelectAccessData	true
clickMe	SelectAccessData

POST Tests

Figure 7Successful Servlet Page

Testing Your Deployment

Testing your integrated deployment is important when integrating any combination of technologies, especially to ensure your configuration is complete and correct. HP recommends that you check your servlet Enforcer plugin/server combination to ensure that you can:

- Start your Policy Validator.
- Start your server.
- Access Enforcer-protected servlets. This ensures that:
  - — For Tomcat and WebSphere deployments, that the servlet Enforcer plugin has initialized and is mapped to the URL pattern correctly. For all deployments, that this also tests whether or not the servlet Enforcer plugin is loading the correct authentication forms.

  - The servlet Enforcer plugin is contacting the Policy Validator(s) you have configured for that plugin and that the level of access is being returned correctly.
  - For the JSP pages, tests that the Java compiler and the Java Virtual Machine are installed and running correctly.
- Test a wide range of servlets and ensure that they function correctly:
  - A servlet without packages.
  - A servlet with packages.
  - A servlet with packages and utility classes.
- Check the Policy Validator's output to determine whether or not the servlet Enforcer plugin is intercepting HTTP requests before the resource the identity has requested is served. Also check that the Policy Validator is processing incoming requests correctly.

Index