HP OpenView Select Access

Network Integration Guide

Software Version: 6.0

for HP-UX, Linux, Solaris, and Windows operating systems

March 2004

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

Legal Notices

Warranty

Hewlett-Packard makes no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

A copy of the specific warranty terms applicable to your Hewlett-Packard product can be obtained from your local Sales and Service Office.

Restricted Rights Legend

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause in DFARS 252.227-7013.

Hewlett-Packard Company
United States of America

Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19(c)(1,2).

Copyright Notices

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

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

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

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

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

Trademark Notices

- Intel® and Pentium® are registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
- Java™ is a US trademark of Sun Microsystems, Inc.
- Linux is a U.S. registered trademark of Linus Torvalds.
- Microsoft®, Windows®, and Windows NT® are U.S. registered trademarks of Microsoft Corporation.
- Oracle® is a registered US trademark of Oracle Corporation, Redwood City, California.
- UNIX® is a registered trademark of The Open Group.

Support

Please visit the HP OpenView Select Access web site at:

http://www.openview.hp.com/products/select/index.html

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

You can also go directly to the HP OpenView support web site at:

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

The support site includes:

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

Contents

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

Chapter 1

About this guide

This *HP OpenView Select Access 6.0 Network Integration Guide* is intended to help you plan and deploy Select Access in your organization, ensuring that it integrates with your existing backoffice systems. An overview of this guide is listed in Table 1.

See the Release Notes (`relnotes.pdf`) on the Select Access installation CD for known installation issues at the time of this release.

Table 1: Guide overview

These chapters...	Cover these topics...
Chapter 2, *Understanding Select Access*	Gives an introduction to Select Access: what it is, what it does, and how it works.
Chapter 3, *Introducing Select Access concepts*	Describes critical concepts that are essential in understanding Select Access.
Chapter 4, *Deploying Select Access on your network*	Summarizes general deployment scenarios: what your integration options are and what things you need to consider.
Chapter 5, *Preconfiguring a directory server*	Provides a summary of steps you need to take with the directory servers Select Access uses.
Chapter 6, *Setting up your Web server*	Lists the things you need to consider when preconfiguring a Web server to optimize it for use with Select Access.
Chapter 7, *Using Select Access personalization information*	Provides an overview of personalization and how to enable this feature on a Select Access-protected network.

Table 1: Guide overview (Continued)

These chapters...	Cover these topics...
Chapter 8, *Enabling single sign-on*	Introduces and describes the process of single sign-on (SSO) across single and multiple domains on your distributed network.
Chapter 9, *Understanding certificate-based authentication*	Offers an introduction to SSL and describes how certificates are used to enable it.
Appendix A, *Integrating the servlet Enforcer plugin with a servlet engine*	Describes how to integrate the Select Access servlet Enforcer plugin into a servlet container.
Appendix B, *Monitoring Select Access operations*	Describes the Select Access features you can use to monitor activities of Select Access components.
Appendix C, *About Policy Validator queries*	Describes Policy Validator queries: what they are, why they are important, and how you can test Select Access using the Query utility.
Appendix D, *Policy Validator stability: setting file descriptor limits*	Describes what you can do to stabilize Policy Validator on your network.

Who is it for?

This guide is intended for individuals or teams responsible for integrating Select Access with existing legacy systems to ensure HP's access control technologies are implemented and deployed successfully.

What does it assume you already know?

This guide assumes a working knowledge of the following:

- *Select Access features*: You should know how to configure the various components before you install them.
- *LDAP directory servers*: You should understand how data is stored in directory servers in order to ensure information in Policy Builder is set up correctly.

- *Web server and plugin technology*: This helps you to understand how different components of Select Access communicate with each other and with your existing infrastructure.
- *XML*: This helps you to understand how a Policy Validator query is constructed and how configuration files are created.

About the Select Access documentation set

The documentation set includes:

- *HP OpenView Select Access 6.0 Release Notes*
- *HP OpenView Select Access 6.0 Installation Guide*
- *HP OpenView Select Access 6.0 Network Integration Guide*
- *HP OpenView Select Access 6.0 Policy Builder Guide*
- *HP OpenView Select Access 6.0 Developer's Tutorial Guide*
- *HP OpenView Select Access 6.0 Developer's Reference Guide*
- *HP OpenView Select Access 6.0 SAML Solution Guide*
- *HP OpenView Select Access third-party Integration Papers*
- Online Help: Setup Tool and Policy Builder

Related references

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 6.0 Installation Guide*, © Copyright 2000-2004 Hewlett-Packard Development Company, L.P. (`installation_guide.pdf`).
- *HP OpenView Select Access 6.0 Policy Builder Guide*, © Copyright 2000-2004 Hewlett-Packard Development Company, L.P. (`policy_builder_guide.pdf`)
- *HP OpenView Select Access 6.0 Developer's Tutorial Guide*, © Copyright 2000-2004 Hewlett-Packard Development Company, L.P. (`developers_tutorial_guide.pdf`)
- *HP OpenView Select Access 6.0 Developer's Reference Guide*, © Copyright 2000-2004 Hewlett-Packard Development Company, L.P. (`developers_reference_guide.pdf`)
- *HP OpenView Select Access 6.0 SAML Solution Guide*, © Copyright 2000-2004 Hewlett-Packard Development Company, L.P. (`saml_solutions_guide.pdf`)

Integration Papers for third party technologies that you can deploy with Select Access are also available on the product CD in the `docs/solutions` folder.

Chapter 2

Understanding Select Access

Select Access is a centralized access management system that provides you with a unified approach to defining authorization policies and securely managing role-based access to on-line resources. It uses a collection of components that integrate with your network, to give you and your partners the ability to capitalize on the potential of extranets, intranets and portals. These components, along with the access policies you set, offer your Web and wireless users a seamless user experience by connecting them to dispersed resources and applications.

What does Select Access do?

Several features of Select Access extend its functionality beyond that of a simple authorization administration tool. It is a complete access management system, offering you a set of features to support your online relationships with your users and your content partners:

- *Supports single sign-on*
- *Enables user profiling*
- *Provides user password and profile management*
- *Delegates administration*
- *Provides an end-to-end auditing system*
- *Automates the discovery and maintenance of corporate resources*

Together, this extended functionality provides a simplified experience for both the end user and those responsible for managing what the user sees and interacts with.

Supports single sign-on

To improve user satisfaction, Select Access incorporates a Web Single Sign-On (SSO) capability. This means users can sign on once to access all permitted resources and have their information stored for future access. Select Access supports transparent navigation between:

- Multiple proprietary domains: For organizations with ownership of multiple Web sites.
- Multiple partnering domains: For on-line business partners, so they can securely share authentication and authorization information across corporate boundaries that have separate:

- user databases
- authorization policies
- access management products

Using SSO means that users do not have to remember multiple passwords or PINs, thereby reducing the amount of help desk support.

Enables user profiling

A user is represented as a user entry that is stored in a directory server. When you create a user entry, you can also define a set of attributes that describe that user, which become part of the user's profile. The values contained in the attribute can be used in two ways:

- *To determine level-of-access with roles*: Role-based access allows you to configure and apply policies automatically, according to the attribute values stored in the user's profile.
- *To determine delivery-of-content*: Select Access exports user attributes and their values as environment variables, so that applications can use the profile information to personalize Web pages and to conduct transactions.

A user's profile dynamically changes as a user conducts transactions with your organization. As attributes in the profile change, so too can the role the user belongs to. For example, a customer's profile may contain his current bank balance, date of last transaction, and current credit limit—any of which can change from moment to moment.

This capability of Select Access makes development of Web applications much easier, because programmers do not have to develop (or maintain) complex directory or database access codes to extract entitlement information about each user.

Provides user password and profile management

Select Access's password and profile management feature makes it easy for users to conduct business and minimize the demand on technical resources that can best be employed elsewhere. This feature includes the following principles:

- *Password administration*: Allows you to set the policies and expiration times for user passwords. Select Access automates reminders and messages. Other administration features include:
  - Profile lockout and re-activation
  - Password history lists
- *Self-servicing*: Allows users to initiate:
  - The definition of new or existing passwords, which are controlled by the password policy you create.
  - The modification of profile data, which is predefined by the attributes you select. Typically, these attributes are the same

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

attributes the user provides when they register with your organization. If the user is already known to you (like an employee or a supplier), you can pre-populate the values for them.

By allowing users to self-manage passwords and profile data, you reduce the amount of help desk support.

Delegates administration

Delegated Administration allows for delegation of both user and policy management, providing more control for decentralized administrators. Select Access's delegation is highly efficient: it supports sub-delegation to multiple tiers of administrators, which mimics real-world organization charts. This decentralized approach to administration:

- Reduces administrative bottlenecks and costs.
- Puts the power to manage users in the hands of those who best understand those users.

Provides an end-to-end auditing system

Select Access can record all access and authorization actions, as well as all policy administrative changes to any number of outputs, such as:

- The HP Secure Audit server
- JDBC-compliant databases
- Local files
- Platform-specific log files
- Email

Of all output choices, the Secure Audit server is the most useful: not only does it collect messages from different components on a distributed network, but it also allows you to digitally-sign all audit entries and ultimately create a report from the outputs collected.

Automates the discovery and maintenance of corporate resources

In order to define and enforce authorization, Select Access must be aware of all the resources on your network, as well as the users who want to access them. Select Access uses the directory server as the central repository for policy data, which includes the resource listing. You can deploy special HTTP/HTTPS-specific plugins to automatically scan any given network, thereby enumerating available services and resources. As services and resources are enumerated by the plugin, it adds them hierarchically in the Policy Builder's Policy Matrix. Unlike other products that require manual data input (where a simple typing error can put the security of resources at risk) Select Access saves administrators' time and improves accuracy.

How does Select Access work?

Select Access delivers the core of its authorization and authentication functionality with the following technical components:

- *Policy Builder*: Allows full or delegated administrators to define the authentication methods and authorization policies with an easy-to-use administration grid.
- *Policy Validator*: Serves the access decision to the Enforcer plugin after it accepts and evaluates the user's access request with the policy information retrieved from the directory server that holds your Policy Store.
- *Enforcer plugin*: Acts as the agent for Select Access on the Web/application server. The Enforcer plugin enforces the outcome of the access request that has been evaluated by the Policy Validator.
- *SAML server*: Handles the logistics of transferring users between your web sites and those of your partners.

These core components form a sophisticated and consistent architecture that easily adapts to any existing network infrastructure. Primarily XML and Java-based, you can readily extend Select Access to meet the needs of future security requirements.

The authentication process

Select Access's authentication and authorization of Web-based or wireless users takes place within a small number of basic steps. Select Access components communicate via XML documents known as queries and responses. XML offers Select Access complete flexibility for data transmission and integration into existing and future applications, whether Web or non-Web based. Select Access's authentication and authorization process follows these steps:

1. A user makes a request to access a resource.
2. The Enforcer plugin passes details of the request to the Policy Validator, including any authentication information provided.
3. The Policy Validator collects user and policy data from the directory and then caches it for future retrieval.
4. Based on this combination of information, the Policy Validator returns a policy decision to the Enforcer plugin, including relevant information to dynamically personalize the user experience.

Other Select Access components

Other Select Access components provide the support system for Select Access's core components:

- *Administration server & Setup Tool*: As a mini Web server, the Administration server is responsible for the configuration of core components and deployment of the Policy Builder applet in a user's browser. The Setup Tool is a client of the Administration server: it is the interface that allows you to quickly set up and deploy Select Access.

- *Secure Audit server*: Collects and manages incoming log messages from Select Access components on a network.

Third-party components Select Access integrates with

Other third-party components that are integral to an effective Select Access solution:

- *Directory server – LDAP v3.0 compliant*: is the foundation of a Select Access-protected system. It acts as the repository of information. Depending on how you have set up your directory system, Select Access can use one or more directory servers to store:
  - A single policy data location
  - One or more user data locations
- *Web/Application/Portal/Provisioning servers*: are third-party technologies that use Select Access as their authorization and access management system. Depending on your server technology, you can use Select Access's native SSO and/or personalization solution rather than use the server's built-in alternative for a more robust solution.

Figure 1 illustrates how Select Access and third-party components interact with each other.

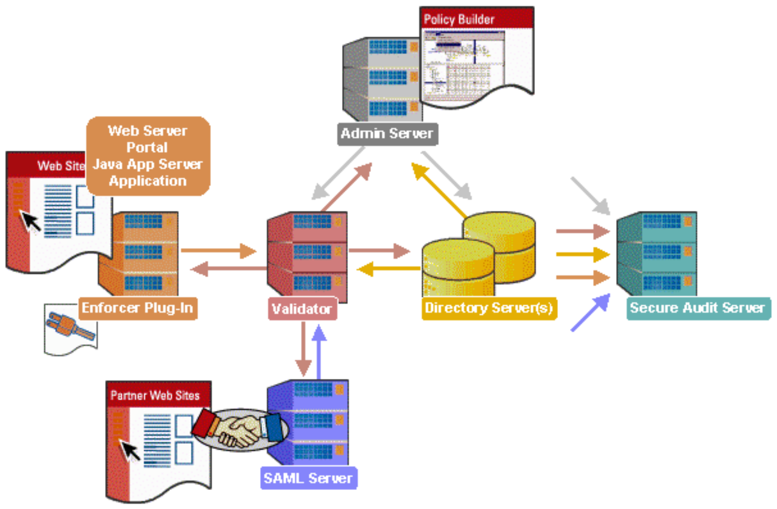


Figure 1: Select Access system architecture

Custom plugins you can customize functionality with

To more efficiently capture your organization's business logic, you can use Select Access's APIs to build custom plugins. Plugins that you can customize functionality with include:

- *Authentication plugins*: A custom Policy Builder authentication plugin allows you to tailor which kinds of authentication methods are available to better meet the needs of your organization. A Policy Builder authentication method plugin allows administrators to use and configure the authentication server for this method via a dialog box. As with the decision point plugin, this dialog box is a property editor that allows security administrators to configure the authentication server.

- *Decision point plugins*: A custom Rule Builder decision point plugin allows you to tailor how rules are built to better meet the needs of your organization. A Rule Builder decision point plugin allows administrators to use and configure the criteria for the decision point via:
  - The icons that represent that decision point on both the toolbar and the rule tree.
  - The dialog box, known as a property editor, that allows security administrators to configure it.
- *Policy Validator decider plugins*: The Validator-specific counterpart of a decision point plugin, the decider plugin allows you to capture the evaluation logic for your custom decision point (described above), so that the Policy Validator can evaluate users based on the information it collects.
- *Resource discovery plugins*: These plugins allow you to customize how resources are scanned on your network.
- *Enforcer plugins*: A new Enforcer plugin allows you to customize the backend application logic by enforcing the decision that the Policy Validator returns to the Enforcer plugin’s query.
- *Additional Web/Application/Portal/Provisioning server specific plugins*: These plugins can be included to handle specific integration details between the third-party technology and Select Access. For example, the Domino server requires a `site_data` plugin if you need to transfer site data between Select Access and Domino.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

Chapter 3

Introducing Select Access concepts

As the number of user populations and network resources grow, and the complexity of corporate partnerships increases, the need for an effective, centralized security management tool that secures content in a heterogeneous environment via policy is of mounting importance. While Select Access offers a user-centric policy-based model that is taking an increasing role in today's portal-driven environments, the features it provides and the business processes it influences require a solid understanding of many networking concepts.

To understand how Select Access's access control solution works in an n-tier Web-based distributed architecture, you first need to understand the concepts outlined in Table 2.

Table 2: Overview of key concepts

This concept...	**For details, see...**
Policy-based access control	*Policy-based access control* on page 11
Cookie domains and session management	*Cookie domains and session management* on page 16
Scalability	*Scalability* on page 17
Load balancing and failover	*Load balancing and failover* on page 18

Policy-based access control

Select Access is a policy-based administration system that a single security administrator, or several delegated security administrators, can control. Policy-based access with Select Access is centrally controlled because all policy data is recorded in a directory server that acts as the Policy Store. As a result, irrespective of the number of administrators you use, you are guaranteed to have a consistent set of data to work from that does not require:

- Additional time or tools to maintain disparate security systems
- Additional complex code to manage the security from the Web content and application themselves

What is an access policy?

Authorization management is defined by an access policy. An access policy is an explicit definition of whether a user is allowed or denied access to a resource. If access is conditional, the policy then also defines the criteria for possible access. Figure 2 shows the basic components of a policy.

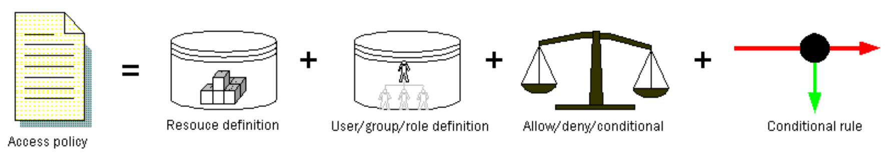


Figure 2: Basic access policy structure

Policy can be manually assigned or automatically inherited across folders, groups, or even entire user locations. Inheritance is used by the Policy Builder to minimize the amount of time it takes to assign policy to thousands or millions of user and resource combinations. For more details on assigning policy, see Chapter 7, *Controlling network access*, in the *HP OpenView Select Access 6.0 Policy Builder Guide*.

How does this compare to policy data?

Do not confuse policy data with the access policy. Policy data includes everything but user data (for example, configured authentication servers and methods, access policies, rules and rule components, and so on). Policy data is written to an entity called the Policy Store, a directory server location configured when you first installed Select Access. For details on policy data, see Chapter 4, *Managing your Policy Data*, in the *HP OpenView Select Access 6.0 Policy Builder Guide*.

User validation and policy enforcement

User validation and policy enforcement are the twin goals of the security model that Select Access provides. Select Access does this with two server components: the Policy Validator (the logic engine) and the Enforcer plugin (the Web server agent).

You can extend the Policy Validator's functionality via customized plugins you create. For details, see the *HP OpenView Select Access 6.0 Developer's Tutorial Guide* and the *HP OpenView Select Access 6.0 Developer's Reference Guide*.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

The process by which a user access request is evaluated by these two components is summarized below:

1. The Enforcer plugin gathers information about a user's incoming access request, and sends what it discovers as an XML query to the Policy Validator.
2. When a query is received, the Policy Validator deciphers the query and determines where additional data look ups need to be performed for the user and resource pair in question.
3. The Policy Validator accesses the required policy information for the user and resource pair from the Policy Store and interprets the policy logic and conditions surrounding the request. This information is cached to improve performance of subsequent queries.
4. If it discovers a conditional policy, the Policy Validator checks and evaluates the rule criteria defined within one or more conditional rules. A decision is reached.
5. The Policy Validator communicates the result of the policy logic to the Enforcer plugin making the request. The Enforcer plugin then enforces the policy decision.

For additional details on this process, as well as a graphical representation of these steps, see *How does the Policy Builder work?* on page 3 of the *HP OpenView Select Access 6.0 Policy Builder Guide*

Query and response structure

Select Access is the only solution in the industry designed from the ground up as an XML-based extensible system.

The query

The Policy Validator receives queries from various Enforcer plugins as XML documents. The Enforcer plugin also encodes queries to the Policy Validator using XML. XML allows for complete extensibility with respect to modifying information in the query structure: you can arbitrarily add new attributes with values to a query.

Select Access ignores any attributes for which it has no use.

Thus, as with the authorization rules, any type of data can be passed to the Policy Validator. This can include:

- Binary objects like X.509 digital certificates
- Complex objects like complete emails
- Simple objects like standard text

For example, an XML query from the Enforcer plugin on behalf of Lance Mountain is shown in Code example 1. This query shows the that the Enforcer plugin's **Query Details** parameter has been set to maximal. Additionally, it shows Lance's personalization details forwarded by the SAML partner, as well as a list of information that describes Steve Smith in the directory server. Notice how data is delimited by the following tag pairs:

- `<PolicyValidatorQuery>` and `</PolicyValidatorQuery>` delimit the entire query and all data contained in it.
- `<PROPERTY NAME="`*`name`*`"></PROPERTY>` delimit the value for the property with `"`*`name`*`"`.
- `<PROPERTYLIST NAME="`*`name`*`">` delimit the value for the property list (in this case a nested property list) with *`"name"`*.

Code example 1: Example Enforcer plugin query to the Policy Validator

```
<PolicyValidatorQuery>
<PROPERTY NAME="service">http://mymenu.foodcompany.com:8070</PROPERTY>
  <PROPERTY NAME="path">/test/auth/submit.cgi</PROPERTY>
  <PROPERTY NAME="dstIP">10.10.10.30</PROPERTY>
  <PROPERTY NAME="srcIP">10.10.10.110</PROPERTY>
  <PROPERTY NAME="dstPort">8070</PROPERTY>
  <PROPERTY NAME="srcPort">4001</PROPERTY>
  <PROPERTY NAME="dstHost">mymenu.foodcompany.com</PROPERTY>
  <PROPERTY NAME="protocol">http</PROPERTY>
  <PROPERTY NAME="srcHost">user_isp.com</PROPERTY>
  <PROPERTY NAME="nonce">
AgAAAAAAPl5SEQAAAAA+XlRqc29sMDAxLmNhLmJhbHRpbW9yZS5jb206OTk5OABwYXNzAGNuPXN0ZXZlIGtvd
HNvcG91bG9zLG91PXN0ZXZlLGRjPWNhLGRjPWJhbHRpbW9yZSxkYz1jb20AAGp79OLYZinTvdZ8UhpWv4CfkX
tmu8885S0AeHA3vX7qrF353ASKBJ5RS2bJz1qCJXGfzK4vQqfVTT0mcE8yIgJQefoFX+GnVV092WaFYtiOe7Q
jfpSqXtaQmShZC1QlRnAYJVwo5Gx5s4B/THU5BgPKZyvUEJnK/pcsb2hOqCOd</PROPERTY>
  <PROPERTYLIST NAME="post_data_list">
    <PROPERTY NAME="fullname">Lance Mountain</PROPERTY>
    <PROPERTY NAME="arrived">10am</PROPERTY>
  </PROPERTYLIST>
  <PROPERTY NAME="native_nonce">enable</PROPERTY>
  <PROPERTY NAME="http_query">hungry=yes&amp;lunch=pizza</PROPERTY>
  <PROPERTYLIST NAME="http_query_list">
    <PROPERTY NAME="hungry">yes</PROPERTY>
    <PROPERTY NAME="lunch">pizza</PROPERTY>
  </PROPERTYLIST>
  <PROPERTY NAME="method">POST</PROPERTY>
  <PROPERTYLIST NAME="http_header_list">
    <PROPERTY NAME="Host">dev01:8070</PROPERTY>
    <PROPERTY NAME="User-Agent">Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US;
rv:1.0.1) Gecko/20020823 Netscape/7.0</PROPERTY>
    <PROPERTY NAME="Accept">
text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,video
/x-mng,image/png,image/jpeg,image/gif;q=0.2,text/css,*/*;q=0.1</PROPERTY>
    <PROPERTY NAME="Accept-Language">en-us, en;q=0.50</PROPERTY>
    <PROPERTY NAME="Accept-Encoding">gzip, deflate, compress;q=0.9</PROPERTY>
    <PROPERTY NAME="Accept-Charset">ISO-8859-1, utf-8;q=0.66, *;q=0.66</PROPERTY>
    <PROPERTY NAME="Keep-Alive">300</PROPERTY>
    <PROPERTY NAME="Connection">keep-alive</PROPERTY>
    <PROPERTY NAME="Referer">http://dev01:8070/test/allow/postfix.html</PROPERTY>
```

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

```
    <PROPERTY NAME="Cookie">
PolicyUser=AgAAAAAAPl5SEQAAAAA+XlRqc29sMDAxLmNhLmJhbHRpbW9yZS5jb206OTk5OABwYXNzAGNuPX
N0ZXZlIGtvdHNvcG91bG9zLG91PXN0ZXZlLGRjPWNhLGRjPWJhbHRpbW9yZSxkYz1jb20AAGp790LYZinTvdZ
8UhpWv4CfkXtmu8885S0AeHA3vX7qrF353ASKBJ5RS2bJz1qCJXGfzK4vQqfVTT0mcE8yIgJQefoFX+GnVV09
2WaFYtiOe7QjfpSqXtaQmShZC1QlRnAYJVwo5Gx5s4B/THU5BgPKZyvUEJnK/pcsb2hOqCOd</PROPERTY>
    <PROPERTY NAME="Content-Type">application/x-www-form-urlencoded</PROPERTY>
    <PROPERTY NAME="Content-Length">54</PROPERTY>
  </PROPERTYLIST>
  <PROPERTY NAME="server">Apache/2.0.40 (Unix) DAV/2</PROPERTY>
  <PROPERTY NAME="queryID">2</PROPERTY>
</PolicyValidatorQuery>
```

The reply

Replies are also formatted as XML. They can contain three things:

- An action telling the Enforcer plugin what to do next. Typical actions are:
  - Allow or Deny
  - Get more information by displaying a form
- Data that must be communicated to an application on the Web server. For example, a Java application might require information to personalize the experience for a user.
- Session information, namely a cookie or nonce that identifies the user on subsequent visits to one or more configured cookie domains. For details on cookies and nonces, see *Understanding nonces and cookies* on page 126.

For example, an XML response from the Policy Validator is shown in Code example 2. Notice how data is delimited by the following tag pairs:

- `<PolicyValidatorReply>` and `</PolicyValidatorReply>` delimit the entire response and all data it contains.
- `<PROPERTY NAME="`*`name`*`"></PROPERTY>` delimit the value for the property with `"`*`name`*`"`.

Code example 2: Example Policy Validator reply to the Enforcer plugin

```
<PolicyValidatorReply>
<PROPERTY NAME="queryID">2</PROPERTY>
  <PROPERTY NAME="authenticated_dn">cn=Lance
Mountain,ou=customer,dc=com,dc=user_isp</PROPERTY>
  <PROPERTYLIST NAME="personalization">
    <PROPERTY NAME="User">lmountain</PROPERTY>
    <PROPERTY NAME="Phone">504%2D2325</PROPERTY>
    <PROPERTY NAME="Fax">504%2D2399</PROPERTY>
    <PROPERTY NAME="DName">
cn%3Dlance%20mountain%2Cou%3Dlance%2Cdc%3Dcom%2Cdc%3Duser_isp%2Cdc</PROPERTY>
    <PROPERTY NAME="Groups">customer%20people</PROPERTY>
  </PROPERTYLIST>
  <PROPERTY NAME="login_time">Thu Feb 27 12:59:45 2003
</PROPERTY>
```

```
  <PROPERTYLIST NAME="authentication_server_types">
    <PROPERTY NAME="authentication_method">password</PROPERTY>
  </PROPERTYLIST>
  <PROPERTY NAME="action">ALLOW</PROPERTY>
</PolicyValidatorReply>
```

What is a Policy Validator cache?

To enhance performance, the Policy Validator employs a cache to reduce the amount of network traffic as well as speed up response time. The Policy Validator cache contains any policy and user data Policy Validator has previously retrieved from the directory server:

- When the Policy Validator receives a policy query for a particular server, it typically caches that policy for future reference.
- For each authenticated user, the Policy Validator caches the user's information.

Due to the fact that runtime conditions can affect the outcome of a given query, the Policy Validator does not cache responses to Enforcer plugin queries. For example, time of day changes can stop a resource from being accessible to a given user after several previous successful accesses.

A cache refresh interval is defined in the Policy Validator's configuration. The Policy Validator therefore checks the directory server and updates the information stored in the cache periodically, so it continually has new information on which to base allow or deny decisions.

The administrator can also flush the cache manually. If the cache expires or is flushed, and a new query comes in for that resource, the cache is again primed with the resource data.

Cookie domains and session management

By manipulating the way cookies are created and used, administrators can force users to explicitly log out from your Web site.

What are cookies and nonces?

Nonces and cookies contain user-specific information. Select Access uses them to help identify users as they navigate through one or more Web servers. They contain some differences, as outlined below:

- *Nonce*: An opaque piece of data the Policy Validator issues to the Enforcer plugin. A nonce can simply be a large random number,

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

or it can contain encrypted or digitally signed data, such as an expiry date.

- *Cookie*: Cookies are issued by the Enforcer plugin and set in the browser. They are the HTTP-specific way of sending the nonce that the Policy Validator issues. The browser sends the cookie in its HTTP headers for each new content request it makes on behalf of the user. This way, the user can be identified without reauthenticating.

For more details, see *Understanding nonces and cookies* on page 126.

Session management

Session management involves administrators being able to force users to explicitly log out of Web sites. For security purposes, it is desirable to allow users to explicitly log out so subsequent users of the shared machine cannot take advantage of the session cookie. For details, see *Forcing user logout* on page 129. It works as follows:

1. When users log into a Web site protected by Select Access, they are issued an HTTP cookie representing their logon session. This cookie typically expires after ten minutes of idle time—a value you can configure for the Policy Validator. For details, seeChapter , *To define the Policy Validator's data encryption settings* in the *HP OpenView Select Access 6.0 Installation Guide*.
2. Users can be logged out in the following two ways:
   – When the cookie expires, the user must log on again if he wishes to access controlled content. The lifetime of the Select Access session cookie also ends when the user completely closes the Web browser.
   – Policy Validator can explicitly log out users by invalidating their session cookie. To log out users, Policy Validator replaces an existing valid HTTP session cookie with an invalid cookie. Invalid cookies cause future connection attempts to fail during the authentication. When Select Access receives an invalid cookie from the Web browser, it forces users to log in again in order to access controlled resources.

Scalability

Select Access is scalable: as the number of users and resources requiring policy increase, the number of Select Access components and servers on which these components rely can also increase accordingly. For details, see *Analyzing your potential for growth* on page 28. Here are some things you can do to meet new growth:

- Install additional Policy Validators to maintain suitable performance levels. Adding more Policy Validators helps reduce the load. There is no limit to the number of additional Policy Validators you can use within a given system. Each query an

Enforcer plugin makes can go to any Policy Validator and get the same answer.

- Delegate administration to remote security administrators. For example, you could have one delegated security administrator for every department of a company that looks after policy creation.
- Install a new Enforcer plugin for each new resource server you add, and add those resources to the Policy Matrix's Resources Tree.
- Distribute directory data so you can scale your directory across multiple directory servers. This is also a good idea in case one directory goes down or in case the load on a given directory server exceeds the CPU capacity of the machine hosting it.

Load balancing and failover

You can increase Select Access's performance and improve its fault tolerance if you configure the following:

- Load balancing: You can replicate Policy Validators so as not to overwhelm with requests any single device. Load is evenly distributed among redundant Policy Validators. For high-volume networks, load balancing is an important aspect for increasing Policy Validator availability as well as increasing performance of your Select Access-protected network. This is because queries can be processed more quickly when Enforcer plugins do not have to wait for a specific Policy Validator to come available. This achieves the following:
  - Increases Policy Validator availability
  - Increases the performance of your Select Access-protected networks

  To deploy Select Access such that it supports load balancing

  - *Configure your Enforcer plugins* so they know to distribute queries among a pool of redundant Policy Validators, rather than use one specific Policy Validator. For details on how to configure Enforcer plugins to support round-robining of Policy Validators, see Chapter 8, *Configuring the Enforcer plugins*, in the *HP OpenView Select Access 6.0 Installation Guide*.
  - *Physically install multiple Policy Validators* on different areas of your back-office network to create a Validator pool. For details on how to configure redundant Policy Validators in a pool, see Chapter 7, *Configuring the Policy Validator*, in the *HP OpenView Select Access 6.0 Installation Guide*.
- Failover: You can improve the fault tolerance of Select Access-protected networks by replicating elements for failover. This ensures that data and components are always available to Select Access clients in the event of a software, hardware, or network fault.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

To deploy Select Access such that it supports failover:

- Replicate directory data to multiple directory servers.
- Configure your Enforcer plugins. Create a Policy Validator pool so Enforcer plugins know how long to wait for a Policy Validator, how many times to retry before the Policy Validator is deemed unavailable to process a query, and where to direct queries when a Policy Validator is considered to be unavailable.

  For details on how to configure Enforcer plugins to support failover, see Chapter 8, *Configuring the Enforcer plugins*, in the *HP OpenView Select Access 6.0 Installation Guide*.
- Physically install and configure multiple Policy Validators. Install Policy Validators on different areas of your back-office network. If you have replicated all or parts of your directory data, you need to further configure Policy Validators to support failover of directory servers.

  For details, see Chapter 7, *Configuring the Policy Validator*, in the *HP OpenView Select Access 6.0 Installation Guide*.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

Chapter 4

Deploying Select Access on your network

Before you begin installing Select Access, allocate some resources to planning your deployment. By taking time up front to understand your network and your needs, you are more likely to deploy an access management system that is as easy to scale as it is to manage. The sections that follow give you the necessary knowledge and advice that reduces any complexity inherent to a component-based product like Select Access.

Issues that affect deployment

All successful deployments of Select Access require some assessment of your current environment. Environmental issues, like those described in Table 3, can have a substantial impact on your deployment of Select Access components. It is important that you analyze your environment to unearth any unexpected integration concerns.

Table 3: Environment deployment issues

This issue...	**Has this impact on deployment...**
Existing security measures that work together to safeguard your organization’s resources and back-office systems from threats of unauthorized access or tampering, denial-of-service, and non-repudiation.	Determine how access management must complement these measures and integrate with them logically. If you require regular auditing or alerts to warn administrators, you need to plan and configure these accordingly. For details, see *Analyzing corporate security* on page 23.
Planned data and component redundancy where replicas work together to prevent system faults and distribute loads evenly across server components on your network.	Decide where and how redundancy works specifically with Select Access components and your directory data. For details, see *Analyzing your redundancy policy* on page 25
Distributed directory data topology where an extremely large number of entries are logically distributed among different servers.	Establish how: • The distribution of data across more than one server impacts your configuration of the Policy Builder. • Select Access supports a directory system that takes advantage of referrals. For details, see *Analyzing your directory topology* on page 27.
Multiple affiliations and partnerships where you plan to integrate sites among your organizations in the near or not-too-distant future.	Requires the addition of a Select Access SAML server to your deployment. You may also need to assess the impact if your partner’s site supports SAML from a different vendor. For details, see *Analyzing your affiliates or partners* on page 28.

Table 3: Environment deployment issues (Continued)

This issue...	**Has this impact on deployment...**
The potential for growth and how you plan to adapt as the number of users and resources requiring policy increases.	Entails looking at your current network resources and determining how readily they scale. For details, see *Analyzing your potential for growth* on page 28.
Existing content servers and any third-party application servers and portal content vendors you have.	Requires that you determine where dedicated and virtual Web servers exist, and how you set up Select Access to work with the latter. It also entails integrating Select Access with these products so that it can protect these vendor resources seamlessly. For details, see *Analyzing your content servers and third-party technologies* on page 29.

Analyzing corporate security

The way you deploy Select Access depends on how you use access management to support your existing security system. Select Access can add a new level of security when shielding your network resources and other enterprise-level back-office systems.

What security systems do you currently use?

You may already have a security system that is well adapted to your needs. Select Access needs to integrate with any of the following security techniques:

- *Authentication schemes*: Select Access currently supports the following authentication schemes: password, registration, certificate, Integrated Windows, NTLM, Kerberos, SecurID, and RADIUS. For each scheme that Select Access supports, you must provide configuration details that allows Select Access to use these methods. For details, see Chapter 6, *Setting up authentication servers*, of the *HP OpenView Select Access 6.0 Policy Builder Guide*. If you have an unsupported authentication scheme, you need to create a plugin for it. For details, see the *HP OpenView Select Access 6.0 Developer's Tutorial Guide* and the *HP OpenView Select Access 6.0 Developer's Reference Guide*.

If you intend to use SecurID as one of your authentications methods on Unix, ensure that the `sdconf.rec` file can be found in `/var/ace/`.

- *Password policies*: Select Access allows you to configure a password policy. If you have an existing corporate policy on passwords you want to set up the Select Access policy so that it closely replicates important criteria of it. For details, see *Configuring password policies* on page 161 of the *HP OpenView Select Access 6.0 Policy Builder Guide*.
- *SSL encryption and PKI systems*: Select Access uses certificate-based SSL encryption to protect the integrity of information as it is communicated among Select Access components, or between the directory servers Select Access relies upon for its user data. There are two ways you can deploy Select Access with SSL:
  - *Using your own CA certificates.* This option is the more difficult of the two options, due to the inherent complexity of different SSL configuration options among Select Access components. If you intend to use your own CA certificates, ensure that you perform a **Custom** setup.
  - *Letting Select Access handle SSL its own way.* This is the recommended method of enabling SSL because it removes all of the complexity of configuring all Select Access components separately. If you do not need to configure SSL, ensure you perform a **Typical** setup.

  For details on setting up SSL, see the component configuration chapters of the *HP OpenView Select Access 6.0 Installation Guide*.

SSL encryption only protects data while it is in transit. If you have concerns over protecting data at its source, you can use digital signatures to sign policy data in the directory server acting as your Policy Store. For details on signing policy data, see *Chapter 4, Managing your Policy Data*, in the *HP OpenView Select Access 6.0 Policy Builder Guide*.

Do you require regular auditing and or reporting of your network activities?

If so, you want to install a Secure Audit server to extend your current corporate auditing standards to include all Select Access components. The Secure Audit server allows you to track runtime messages of all Select Access components—especially when they are installed on a distributed network. It audits and collects log files Select Access clients generate so you can check:

- The efficiency of all components

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

- Problems they might experience at runtime
- Conceivable or actual security threats or breaches

The Secure Audit server is not typically required in a small Select Access deployment. Because components are typically installed over a small, localized number of machines, collecting logs is a simple task that does not warrant a separate server.

If you use a JDBC-compliant database, you need to configure it correctly before the Secure Audit Server can log to it.

First, you need to enable database reporting in the Administration server setup wizard (this option is only available when you perform a **Custom** installation). For details, see *To configure database reporting* on page 82, in the *HP OpenView Select Access 6.0 Installation Guide*.

Second, you need to run the corresponding SQL script installed with Select Access. For details, see *Configuring a database* on page 103 in the *HP OpenView Select Access 6.0 Installation Guide*.

To make the auditing of Select Access components more targeted, you can also create reports from the Secure Audit server's database or file output, as well as configure email alerts for more severe events that require immediate notification. For details on creating alerts and generating reports, see *The alert notification decision point* on page 136 and *Creating and viewing a report with the Report Viewer* on page 145 in the *HP OpenView Select Access 6.0 Policy Builder Guide* respectively.

Analyzing your redundancy policy

Redundant systems can have different goals, ranging from increasing performance to increasing fault tolerance. Different Select Access components support some, if not all, of these goals via a system of replication. There are two different meanings of replication, depending on the component:

- *Replication of directory data:* is a mechanism that automatically copies data from a master directory server to a slave server.
- *Replication of software components:* is a mechanism that requires you to manually install duplicate Select Access components on different areas of your network.

How do you increase the performance of Select Access components?

You can increase the performance of components by configuring replicated Policy Validators for round-robining. Round-robining is a system whereby load is evenly distributed among redundant Policy Validators. For high volume networks, round-robining is an important aspect for increasing Policy Validator availability as well as increasing performance of your Select Access-protected network. This is because queries can be processed more quickly when Enforcer plugins do not have to wait for a specific Policy Validator to become available.

To deploy Select Access such that it supports load balancing:

- *Configure your Enforcer plugins* so they know to distribute queries among a group of redundant Policy Validators, rather than use one specific Policy Validator. For details on how to configure Enforcer plugins to support round-robining of Policy Validators, see Chapter 8, *Configuring the Enforcer plugins*, in the *HP OpenView Select Access 6.0 Installation Guide*.
- *Install multiple Policy Validators* on different areas of your back-office network to create a list of Validators that can share network load.

⚠ Always install all Policy Validators on the same network as your other back-office systems. The firewall you use to protect these systems needs to also protect Policy Validators from attack.

⚠ If you have a group of Policy Validators, the RSA algorithm-based key needed to validate Select Access cookies must be shared with other Policy Validators. As a result, it must be published to the Policy Store. This ensures that users do not have to reauthenticate because the required key to validate cookies was not published.

To share the key, ensure that you check the **Share key with other Validators** box in the Policy Validator **Secure User Credentials** setup screen. Note that this screen only appears when you do a **Custom** setup.

For details on how to configure redundant Policy Validators, see Chapter 7, *Configuring the Policy Validator*, in the *HP OpenView Select Access 6.0 Installation Guide*.

How do you increase the fault tolerance of Select Access components?

You can improve the fault tolerance of Select Access-protected networks by replicating the following elements for failover:

- *Replicate directory data to multiple directory servers.* This ensures data is always available to Select Access clients in the event of a software and/or hardware and/or network fault. For details on how to integerate this functionality with your Select Access deployment, see Chapter 5, *Preconfiguring a directory server*, in the *HP OpenView Select Access 6.0 Installation Guide*.
- *Configure your Enforcer plugins.* Create a list of Policy Validators, so Enforcer plugins know:

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

- – How long to wait for a Policy Validator or how many times to retry it before it is deemed unavailable to process a query
- – Where to direct queries if a Policy Validator is considered to be unavailable

For details on how to configure Enforcer plugins to support failover, see Chapter 8, *Configuring the Enforcer plugins*, in the *HP OpenView Select Access 6.0 Installation Guide*.

- *Physically configure multiple Policy Validators*, so they know to which directory servers to failover to. Install Policy Validators on different areas of your back-office network. If you have replicated all or parts of your directory data, you need to further configure Policy Validators to support failover of directory servers.

⚠ Always install all Policy Validators on the same network as your other back-office systems. The firewall you use to protect these systems needs to also protect Policy Validators from attack.

For details, see Chapter 7, *Configuring the Policy Validator*, in the *HP OpenView Select Access 6.0 Installation Guide*.

Analyzing your directory topology

Most directory servers allow you to design a distributed directory in which user data is spread across multiple physical directory servers. The benefit of distributing user data among distributed servers is that you improve the performance of the directory system as well as other client applications that make use of that data.

How does a distributed directory affect Select Access?

If you have a distributed directory topology, you likely have set up a network of referrals among these different servers. Referrals allow you to hide the functional details of your distribution, by allowing servers to redirect queries to the appropriate data location. This process of referrals makes your directory topology seem as if it were centralized on one machine.

A distributed topology affects your deployment in that it requires you to check the type of referrals you use. Typically, there are two: default referrals and smart referrals. Select Access does not support default referrals. If your distributed directory system uses default referrals, you need to change the mechanics of your referral system to use smart referrals only. If you use smart referrals, ensure that the referral syntax is compatible with Select Access's referral support.

For details on deploying Select Access with a directory system that uses referrals, see Chapter 5, *Preconfiguring a directory server*, in the *HP OpenView Select Access 6.0 Installation Guide*.

Analyzing your affiliates or partners

Select Access supports a protocol that creates a seamless user experience, which avoids the necessity for reauthentication when a user tries to access resources that are hosted by an affiliate organization. This protocol, known as Security Assertions Markup Language (SAML), creates an interoperable mechanism for passing credentials and other related information among servers belonging to partnering organizations—even if they have their own authentication and authorization systems.

How do I take advantage of SAML with my Select Access deployment?

There are two main issues to keep in mind:

- You and your SAML partners must:
  - Exchange unique identifiers, connection parameters, and assertion processing details.
  - Agree on what attribute namespaces you require.
  - Determine which servers are responsible for sending and/or receiving authenticated users to partner sites.

  Without knowing these details in advance, you are not able to configure your SAML server.
- The Select Access SAML server is a separate process. Consequently, it can run on either the same host machine as the Validator and/or the Web server or on a different host altogether. The SAML component communicates directly with the Validator, so that it can send and retrieve the information it needs and it communicates with the other SAML servers on the network with which it has a relationship.

Analyzing your potential for growth

As the number of users and resources on your network increases, so too does the number of Select Access components and servers on which these components rely. Select Access's componentry allows your access management deployment to scale according to the change in volume.

How do I roll out new components in stages?

Depending on your current size, there can be varying combinations of new components you need to:

- *Monitor your loads closely*. If you start reaching upwards of 75% of your capacity, consider installing additional Policy Validators to maintain suitable performance levels.
- *Find a suitable balance between the number of users and the number of security administrators*. As your organization grows, you may find the need to delegate administration to remote security administrators.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

- *Install a new Enforcer plugin* for each new resource server you add, and add those resources to the Policy Matrix's Resource Tree. Ensure you make this part of your deployment procedure for new servers. For details, see Chapter 8, *Configuring the Enforcer plugins*, in the *HP OpenView Select Access 6.0 Installation Guide*.
- *Distribute directory data* so you can scale your directory among multiple directory servers. A distributed directory can adapt more quickly when user numbers change over time. For each new branch of directory data you add, you also need to configure the corresponding branch of the Users Tree in the Policy Matrix. For details, see Chapter 3, *Building your Users and Resources Trees*, in the *HP OpenView Select Access 6.0 Policy Builder Guide*.

Analyzing your content servers and third-party technologies

Web servers and other third-party content servers help your organization to thrive on the Web. Your content system may include any combination of:

- Dedicated Web servers
- Virtual Web servers
- Application servers
- Portal content and portal servers

How do third-party application or portal vendors affect my deployment?

Select Access supports certain application servers and portal vendors. If Select Access supports a vendor, you can integrate Select Access with this third-party technology to ensure that its resources are Select Access-protected. To determine which third-party products you can integrate with, check `<install_path>/docs/solutions` for the list of technology-specific integration documents available to you.

Deployment scenarios

How you deploy your version of a Select Access access management solution depends upon the size and nature of your organization. Table 4, provides two general examples of how Select Access can be configured. You can extend these examples into your specific environment and use them as a starting point for developing your own access management deployment plan.

Table 4: Example Select Access deployment scenarios

Installing Select Access on...	**For details, see...**
A *single machine* that has most Select Access components installed on it, with the exception of the Enforcer plugins and a directory server. This example: • Is the simplest of all Select Access deployment scenarios • Is typically used by mid-size organizations • Takes advantage of only some of Select Access's features	*Centrally located deployment by a mid-sized manufacturer* on page 30
Several geographically dispersed machines that have a single Select Access component installed on them, and the topology of the directory system is replicated and globally dispersed. This example: • Is the most advanced of all Select Access deployment scenarios • Is typically used by large multi-national enterprises • Takes full advantage of Select Access's features and componentry	*Fully distributed deployment by a multi-national enterprise* on page 33

Centrally located deployment by a mid-sized manufacturer

Ball Brothers Manufacturing (BBM), a ball bearings manufacturer, is a mid-size company that consists of 600 employees, with a single office in Flint, Michigan. BBM recently purchased a single instance of iPlanet v 5.0 directory server, and wanted to extend it by purchasing Select Access to manage access of network resources, much of which is of a highly sensitive nature.

Other significant details include:

- All of BBM's user data is centrally managed by a single directory administrator, who also takes on the role of security administrator as well. If possible, the security administrator would like to work from her home office, due to extensive commuting times.
- BBM has little knowledge of PKI, and has no plans to invest in this kind of infrastructure.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

- BBM recently launched an Internet site that is intended to facilitate work processes for their suppliers and simplify ordering for customers.
- BBM has a limited IT budget so existing hosts must be used wisely.
- BBM does not allow anonymous access and requires that new and unknown users register to the directory server. Known users must authenticate via password.

Their Select Access solution

Based on their business and IT environment, BBM's best deployment is one that is small and centrally located. Therefore, deployment of BBM's components are installed among a few host computers—hosts that also host other applications. This solution requires minimal network topology and firewall configuration planning, and does not require all of Select Access's components.

In this scenario, BBM likely has:

- A single firewall that divides its network into public and private zones
- A single host that includes:
  - A directory server that holds all user and policy data
  - The Administration Server
  - The Policy Validator
- Another computer that hosts only an Enforcer-protected Web server

Location of Select Access components

Because BBM is a mid-size organization, it is not likely to require all of the robust features and components included with Select Access. Like most organizations of this nature, BBM has divided its network into two zones. Table 5 summarizes on which network zone it plans to install Select Access components.

Table 5: Deploying Select Access on a localized network

This network zone...	Hosts this component...
DMZ network	• A single Enforcer-protected Web server accessed over the extranet
Internal network	• A single Policy Validator • The Administration server/Policy Builder • The directory server that holds all user, policy, and configuration information Select Access uses

Figure 3 illustrates the topology of this deployment.

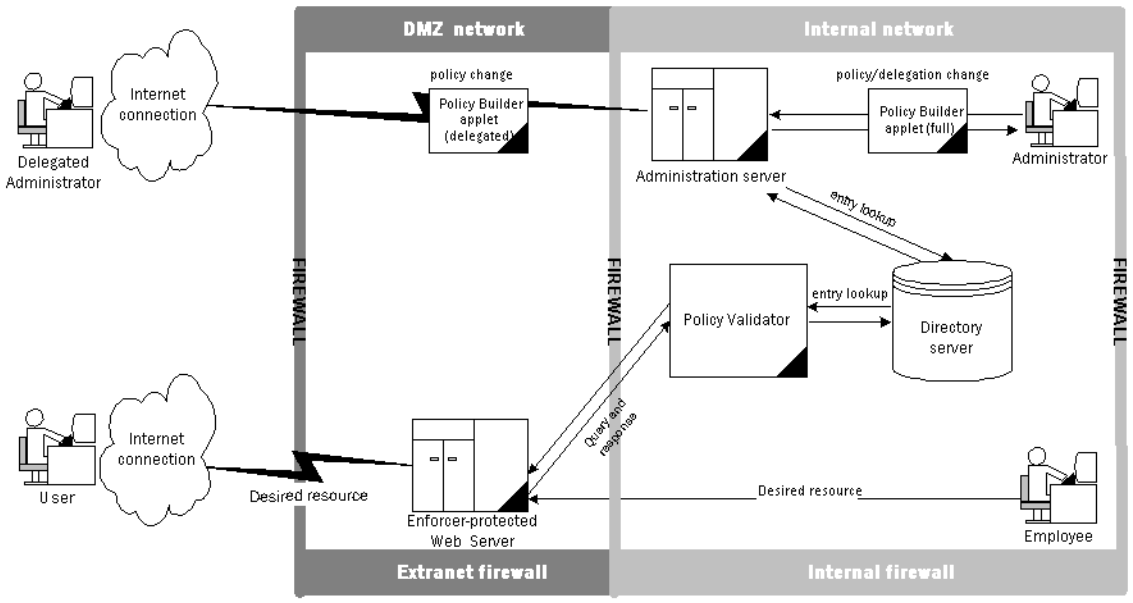


Figure 3: Localized Select Access deployment

Component configuration

After installing each component in the appropriate physical location, BBM accepts all of Select Access's setup defaults by performing a **Typical** setup for all components. A Typical setup is generally performed when you do not need to customize configuration or enable special features. Therefore, the only configuration values BBM provides are those for host and server names, and locations.

Additional features used

After all components are up and running BBM still needs to configure Select Access to take advantage of the following features:

- Password and registration authentication support, which also includes taking the corresponding form templates that were installed with Select Access and customizing them
- User accounts so suppliers and/or customers can update their own profile info
- Password policies to match BBM's existing corporate policy
- The delegated administration applet so the full administrator can determine and set delegated administration policies that allow delegated users to administrate users and/or policies remotely

Fully distributed deployment by a multi-national enterprise

Magellan Financial International (MFI), a financial services provider, is a large multi-national enterprise that consists of thousands of employees world-wide, with regional offices in North America, Europe, and South East Asia. MFI has legacy back-office systems on different platforms that include Windows 2000, Linux, and Solaris. To help them manage their employees, as well as their vast network of agents, brokers, and clients, MFI purchased several licences for the Siemens DirX directory server a year ago. Since then, it has successfully:

- Logically distributed directory entries among all regions
- Used replication and referrals to ensure 100% availability and improve throughput of data

Other significant details include:

- Due to the amount of user data that must be managed across multiple locations, MFI has recently created a new role of security administrator. There is one senior security adminstrator to manage the process on a global scale and several security administrators to whom access management is delegated.
- MFI has recently purchased UniCERT PKI to further shield their systems from attack, and to facilitate the authentication of their users.
- MFI has recently launched an extranet that is intended to improve relationships with their network of agents and brokers and give them the knowledge they need to give sound financial advice in their region. The extranet also allows clients to view or modify their portfolio as needed.
- MFI has a large IT budget. One of the CEOs mandates is to continue to use any technology available to:
  - Reduce overhead costs
  - Build online relationships with agents, brokers, and clients

- MFI does allow anonymous access. However, a more robust online experience requires that new and unknown users authenticate to the system via registration, password-based authentication, or, for more sensitive resources, certificate-based authentication. The executive team is also using SecurID to access legal, acquisition, and financial resources that are only available on a "need-to-know" basis.
- MFI has also started negotiations to affiliate with a leading insurance provider so it can offer additional services to its clients. It plans to close the deal within a few months, but is still unsure how to avoid having the user reauthenticate as she moves among the two corporate sites.
- MFI's VP of IT also requires that all major security schemes be audited to watch for possible security issues, and to be notified when any security application has problems at run time.

Their Select Access solution

Based on their business and IT environment, MFI's deployment is best served by a large scale, full-featured configuration. Therefore, deployment of components would be distributed among multiple machines. This solution requires a lot of network topology integration and firewall configuration planning, which needs to take full advantage of Select Access's componentry.

In this scenario, MFI likely has:

- Firewalls that control traffic between network zones and components. You need a firewall for your extranet and Internet traffic, and a firewall for your intranet and back-office traffic. For details, see *Location of Select Access components* on page 35.

If MFI had a NAT firewall, they would also have to map network addresses for Enforcer plugin and Policy Validators so they could communicate over this firewall successfully. They can configure this mapping when setting up their Enforcer plugins. For more information, see the *HP OpenView Select Access 6.0 Installation Guide*.

- Multiple directory servers:
  - One for each regional office, which holds the user data for that region. These directory servers also use a system of smart referrals to redirect third-party application clients to the corresponding database server location.
  - One for the Select Access Administration server and Policy Store, which holds the configuration information and policy data.
  - Several replicas of the original servers to protect against possible server or network faults.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

- Several Enforcer-protected Web servers hosted on their own machines and on different platforms.
- An administration server to serve up both the Policy Builder applet and the Delegated Administration applet. This allows administrators who have had administration privileges delegated to them to remotely configure access policies to their segment of users.
- A validator pool, consisting of three Policy Validators. A computer in each regional office hosts these Policy Validators.
- A SAML server, which allows MFI to exchange user credentials with partnering sites to offer their users a seamless Web experience.
- A Secure Audit server to coordinate the logging activity among Select Access components, and simplify the generation of reports by collecting data from geographically dispersed components. The Secure Audit server is configured to output to a JDBC-compliant database that has been set up to work with Select Access.

Location of Select Access components

Because MFI is a multi-national organization, deployment options are quite advanced. Like most organizations of this nature, MFI has divided their network into three zones. Table 6 shows that there are three network zones that fully distributed organizations employ.

Table 6: Deploying Select Access on a distributed network

This network zone...	Hosts this component...
DMZ network	• Any Enforcer-protected Web servers accessed over the Internet • A SAML server
Back-end network	• Policy Validators • The directory server that holds all user, policy, and configuration information used by Select Access • The Administration Server (and Policy Builder applet)
Internal network	• Any Enforcer-protected servers that host internal systems or applications

Figure 4 illustrates the topology of this deployment.

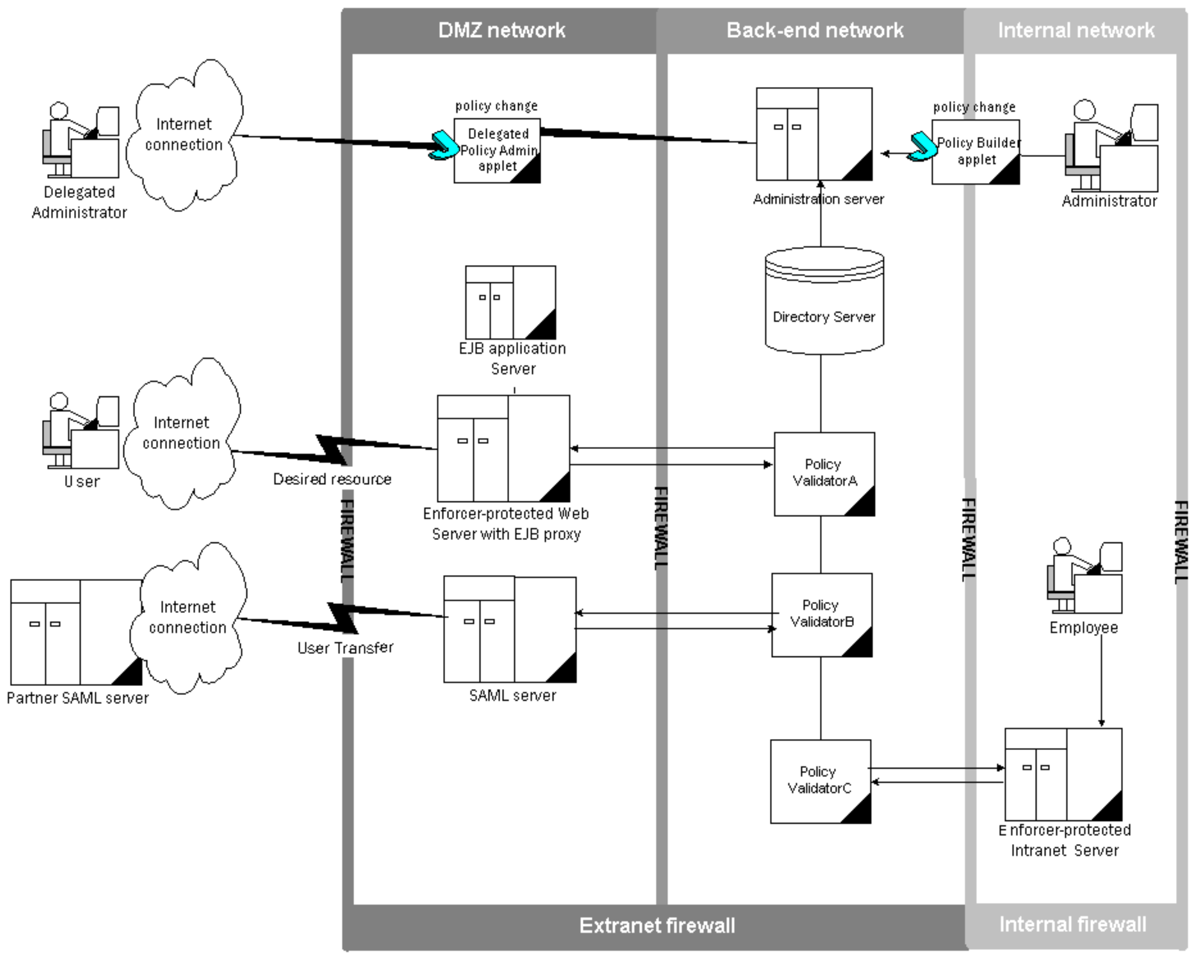


Figure 4: Fully distributed Select Access deployment

Component configuration

Because of the complexities of this deployment, MFI needs to customize its configuration of Select Access components. This allows them to tweak their installation to allow for their unique needs. It would not perform a **Typical** setup on any of its components. Instead, all components need to be customized.

Additional features used

After all components are up and running MFI still needs to configure Select Access to take advantage of the following features:

- Password, certificate, and registration authentication support, which also includes taking the corresponding forms templates that were installed with Select Access and customizing them
- User accounts so clients can update own profile info
- Password policies to match MFI's existing corporate policy
- Take full advantage of the Log Viewer to create reports from the JDBC-compliant log that the Secure Audit server outputs to

Chapter 5

Preconfiguring a directory server

Select Access uses a Lightweight Directory Access Protocol (LDAP) v.3-compliant directory server for searching and storing user information. By using standards-compliant LDAP servers and meta directories for access to legacy user data stores, Select Access enables you to synchronize information easily across a globally dispersed network—including those with multiple delegated administrators.

Supported directory servers

To make the adoption of Select Access as easy as possible, HP has included support for several LDAP v.3-compliant directory servers. These directory severs are documented in *Supported LDAP directory servers* on page 14.

Before you install and configure Select Access, prepare your directory server and the data it contains to ensure it is compatible with Select Access components. Depending on your directory requirements, you can store your user data on a single directory server or across multiple servers.

If your directory system has been designed to support replication and/or referrals, you need to take additional steps to configure Select Access components correctly for this configuration. For an overview of using replication and referrals with Select Access, see *Setting up replication and referrals* on page 67.

To see a list of which characters are supported by specific directory servers only, see Appendix B, *Invalid characters* in the *HP OpenView Select Access 6.0 Policy Builder Guide*.

Password authentication differences among directory servers

Select Access takes advantage of two main authentication routines:

- Validator-based authentication is the authentication routine of choice because it is faster. This authentication routine allows Select Access to read the password directly, thereby allowing the Policy Validator to perform the authentication.
- LDAP-based authentication is used as an alternative to Validator-based authentication when a directory server does not support the latter. This authentication routine lets Select Access bind as the user before being allowed to read the password. In this case, authentication occurs outside the Policy Validator.

Certain directory servers do not allow Select Access to read the user's password. When this happens, the password seemingly appears undefined, despite being present in the directory server.

Table 7 summarizes the subtleties among supported directory servers. Review this table and ensure you understand the implication of these differences.

Table 7: Password authentication summary

Directory server	Authentication routine	Read password	Set and manage passwords
Sun ONE 5.1	Validator	Yes	Yes
Netscape 6.01	Validator	Yes	Yes
Novell eDirectory 8.5.1	LDAP	No	Yes
Siemens DirX 6.0A10	Validator	Yes	Yes
Critical Path Directory 4.0	LDAP	No	Yes
Microsoft Active Directory 5.0	LDAP	No	Via SSL only [1]
Oracle Internet Directory 3.0.1.1	Validator	Yes	Yes
eTrust Directory Server	Validator	Yes	Yes

1. See the *HP OpenView Select Access 6.0 Release Notes* for details.

Select Access and directory server user schemas

Because Select Access requires tight integration with directory servers, schemas for some require specific updates. This ensures that policy and user 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. Depending on the directory server, some schemas require manual modification before

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

Select Access can map to it. In others, Select Access maps to the schema automatically.

If your directory server requires a schema update, you must do this before you install and run Select Access.

Table 8 outlines the procedure for modifying schemas manually and lists the directory servers that require manual intervention. These steps are only required if this is your first time installing Select Access.

If you are upgrading from a previous version, there are separate procedures you may need to follow. For details, see *Updating certain directory servers' schemas* on page 63.

Table 8: Modifying schemas

This step...	For details, see...
1. Modify your directory server's schema. If you do not see your directory server listed in the adjacent column, assume that there are no Select Access-specific schema modifications that are required, and that the server's schema uploads transparently.	*The Sun ONE and Netscape schemas* on page 40 OR *The Siemens DirX schema for policy and user data* on page 40 OR *The Siemens DirX schema for user data only* on page 46 OR *The CP Directory schema for v4.0* on page 50 OR *The CP Directory schema for v4.1* on page 52 OR *The Microsoft Active Directory schema* on page 53 OR *The Oracle Internet Directory schema* on page 57 OR *The eTrust Directory server schema* on page 59

Table 8: Modifying schemas (Continued)

This step...	For details, see...
	The Oracle Internet Directory schema on page 57 OR *The eTrust Directory server schema* on page 59
2. After Select Access has successfully uploaded your directory server's schema, tune the schema by indexing attributes as required.	*Tuning directory servers by indexing attributes* on page 66

The Sun ONE and Netscape schemas

If you are using Sun ONE v5.1 or Netscape v6.01, you need to enable the `uid uniqueness` plugin for each LDAP server, including the ones used for referrals. Other than enabling this plugin, no specific manual schema modifications are required.

To enable the uid uniqueness plugin

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 **Administration Server**, and click the **Open** button. The **Administration Server** window appears.
4. Click the **Configuration** tab and expand the **Plugins** tree entry.
5. Click the **uid uniqueness** entry. A configuration screen appears on the right hand side.
6. Select **Enable plug-in** and then click **Save**.

The Siemens DirX schema for policy and user data

If you are using Siemens DirX as a Policy Store as well as one or more of your user data locations, you must manually upload the schema. Table 9 outlines the steps you must take before you can upload DirX's schema on Select Access.

Table 9: DirX as a Policy Store and user data location

This step...	For details, see...
1. Copy Select Access-specific files to DirX's working folder.	*To relocate Select Access files to the corresponding DirX location* on page 41
2. Modify the global variables defined in the `GlobalVar.tcl` file.	*To modify the global variables* on page 42

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

Table 9: DirX as a Policy Store and user data location

This step...	**For details, see...**
3. If you are modifying the user schema for a directory server on Windows, rebuild your directory server and update the user schema by running `setup.bat`. **Note:** If running the `setup.bat` file fails to rebuild the entire directory server and update the Select Access schema, then you need to set up DirX manually.	*To run setup automatically on Windows* on page 43 OR *To set up DirX manually on Windows* on page 44
4. If you are modifying the user schema for a directory server on Unix, you need to perform steps manually to rebuild the directory server and to update the schema	*To update the schema for DirX on Unix* on page 44

To relocate Select Access files to the corresponding DirX location

1. Locate the files you need. By default, all schema-related files are installed to `<install_path>/schema/Siemens-DirX/`.
2. Copy the files to the corresponding DirX folder. Table 10 summarizes which files are needed when DirX acts as a Policy Store as well as a user data location.

Some of the files are platform-specific. You only need to copy files that apply to the platform your directory server is running on.

Table 10: Files required when using DirX as Policy Store and user data location

This Select Access file...	Goes to this DirX location...
• `dirxabbr-ext.SelectAccess`: Is a Select Access-related extension of the `dirxabbr` file. It maps object identifiers of Select Access's attributes, object classes, and name forms to names and abbreviations. The objects themselves are later created by scripts.	`<install_path>\Client\Conf`
• `globalVar.tcl`: Contains global variable definitions. You need to modify this file to suit your network environment. For details, see *To modify the global variables* on page 42. • `l-bind.cp`: Performs a DirX administrator LDAP bind. • `schema_ext_for_SelectAccess.adm`: Extends the global schema with elements defined for Select Access. • `setup.bat`: Is a batch file that sets up Select Access using an empty DirX database. In other words, this file rebuilds the entire directory server and updates the Select Access schema (for Windows only). • `subschema_ext_for_SelectAccess.cp`: Provides the subschema extension of the DSA schema. • `accessControl_SelectAccess.cp`: Enables access control for Select Access administrators (for Windows only). • `bind.tcl`: Performs a DirX administrator bind (for Unix only). • `create_SelectAccess_admin.cp`: Creates a Select Access administrator entry (for Windows only). • `Extend_LDAP_Root.cp`: Indicates to Select Access that the directory is a Siemens one.	`<install_path>\scripts\security\SelectAccess`

To modify the global variables

1. Open `globalvar.tcl`. This file is one of the Select Access-specific files you copied in *To relocate Select Access files to the corresponding DirX location* on page 41.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

2. Define values for the variables it contains to customize the file for your organization. Table 11 explains the variables contained in this file.

Ensure the `country` and `organization` variables are identical to the ones in your existing DSA or you are not able to log into the database.

The `country` and `organization` variables combine to form a DN in the LDAP database.

Table 11: Global variables and their new values

For this variable...	Use this value...
`country`	Your country's Internet domain. For example, for Germany, type `DE`.
`organization`	Your organization's name. For example, `sample.com`.
`name`	The directory server's administrator name or user ID. For example, `JDoe`.
`defaultPW`	A default or customized password. This variable is case sensitive.
`saAdminName`	Select Access's security administrator's name or user ID. For example, `saAdmin`.
`saAdminPw`	A default or customized password. This variable is case sensitive. For example, `W3wizard`.
`ldapPort`	The default LDAP port number. For example, `389`. Change the port number if the default port is already being used by another application.

The login name, which is case insensitive, is made up of the `organization` and `name` variables.

To run setup automatically on Windows

1. Run the scripts from the following location: `<install_path>\schema\Siemens-DirX\setup.bat`. The `setup.log` file then appears in your default text editor.
2. Ensure that the setup was successful by searching `setup.log` for the word `FAILED`.
   - If you do *not* find this word in the log file, your setup was successful.

- If you *do* find this word, automatic setup has failed. In this case, you need to set up DirX manually. For details, see *To set up DirX manually on Windows* on page 44.

3. Close the `setup.log` file.

To set up DirX manually on Windows

1. Ensure that Siemens DirX is running and that your database is empty. If it is not, do the following:
   a. Stop the Siemens DirX service from running.
   b. Run the following files in the order they appear:
      `<DirX_install_path>\bin\mkbootsch`
      `<DirX_install_path>\bin\mkdbconf`
      `<DirX_install_path>\bin\mkdb`
   c. Restart the Siemens DirX service.
2. To build the directory server from scratch, run the `dirxadm` application with the following scripts:
   - `source bootstrap_DSA.adm`
   - `source ldap_admin.adm`
3. Exit `dirxadm`.
4. To set up the standalone directory system agent (DSA) using the variables you defined in `GlobalVar.tcl`, run the `dirxcp` application from the same location with the following files:
   - `source initialize_DSA.cp`
   - `source load_DSA.cp`

   Select Access needs a standalone DSA so its components can read and write to the DirX directory server.
5. Exit `dirxcp`.
6. To extend the DSA schema with the Select Access-specific attributes, object classes, and forms, rerun the `dirxadm` application a second time with this new file:
   `source schema_ext_for_SelectAccess.adm`
7. To add the DirX schema to Select Access, rerun the `dirxcp` application with the following Select Access-specific file:
   `source subschema_ext_for_SelectAccess.cp`
8. To define a Select Access administrator and set the correct access rights for this administrator, rerun the `dirxcp` application with these files:
   - `source create_SelectAccess_admin.cp`
   - `source accessControl_SelectAccess.cp`
9. Exit from `dirxcp`.

To update the schema for DirX on Unix

1. Ensure that Siemens DirX is running and that your database is empty. If it is not, do the following:

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

   a. Stop the Siemens DirX service from running.
   b. Run the following files in the order they appear:
      `<DirX_install_path>/bin/mkbootsch`
      `<DirX_install_path>/bin/mkdbconf`
      `<DirX_install_path>/bin/mkdb`
   c. Restart the Siemens DirX service.
2. Log into the DirX root account on your system.
3. Copy the `/mnt/cdrom/<DirX_scripts>/*` scripts to the following location:
   `<install_path>\schema\Siemens-DirX\`
4. To build the directory server from scratch, run the `dirxadm` application with the following scripts:
   - `bind <admin_login>`

     where *<admin_login>* is your method of binding the administrator to DirX.
   - `source bootstrap_DSA.adm`
   - `source ldap_admin.adm`
5. Exit from `dirxadm`.
6. To set up the standalone directory system agent (DSA) using the variables you defined in `GlobalVar.tcl`, run the `dirxcp` application from the same location with the following files:
   - `source initialize_DSA.cp`
   - `source load_DSA.cp`

   Select Access needs a standalone DSA so its components can read and write to the DirX directory server.
7. Exit from `dirxcp`.
8. To create the UniCERT schema extensions, perform the following substeps; otherwise, skip to step 9.
   a. Change to the `~/scripts/security/UniCERT` directory.
   b. Run the following scripts from the `dirxcp` application:
      ```
      source schema_mod.cp
      source load_add_DSA.cp
      source access_add.cp
      ```
   c. Exit from `dirxcp`.
9. Change to the `~/scripts/security/SelectAccess` directory.
10. To extend the DSA schema with the Select Access-specific attributes, object classes, and forms, rerun the `dirxadm` application a second time with this new file:
    `source schema_ext_for_SelectAccess.adm`
11. To add the DirX schema to Select Access, rerun the `dirxcp` application with the following Select Access-specific file:
    `source subschema_ext_for_SelectAccess.cp`

12. To define a Select Access administrator and set the correct access rights for this administrator, rerun the `dirxcp` application with these files:
    - `source create_SelectAccess_admin.cp`
    - `source accessControl_SelectAccess.cp`
13. Exit from `dirxcp`.
14. Rerun the `dirxadm` application with the following Select Access-specific file:
    `source bind.tcl`
15. Stop and restart DirX.

The Siemens DirX schema for user data only

If you are using Siemens DirX as a user location only, this directory server requires manual intervention in this case. The files you need to setup DirX as a user location only are listed in Table 12.

Table 12: Files required when using DirX as a user location only

This Select Access file...	Does this...
`dirxabbr-VN-VV.SelectAccess`	Is a Select Access-related extension of the `dirxabbr` file. It contains the attributes needed to allow the Administration server to connect to your server.
`bind.tcl`	Performs a DirX administrator bind (for Unix only).
`globalVar.tcl`:	Contains global variable definitions. You need to modify this file to suit your network environment. For details, see *To modify the global variables* below.
`Extend_LDAP_Root.cp`	A file that indicates to Select Access components that the directory is a Siemens directory.
`add_vendor_attributes.adm`	Extends the global schema by adding variables to the `ldapRootDSE` object class.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

Table 12: Files required when using DirX as a user location only

This Select Access file...	Does this...
`add_goun_nf.adm`	A script that adds the `groupOfUniqueNames` object class.
`goun_rule.cp`	A script that allows parent entries to use the `groupOfUniqueNames` object class, as well as to contain a subentry that uses the `organizationalUnit` object class. Siemens does not allow this by default.

Table 13 outlines the steps you must take, before you use DirX as a user location with Select Access.

Table 13: DirX as a user location only

This step...	For details, see...
1. .Move and rename the `dirxabbr-VN-VV.SelectAccess` file.	*To relocate Select Access's attribute file* on page 48
2. Stop DirX.	DirX's documentation
3. Modify the global variables defined in the `GlobalVar.tcl` file	*To modify the global variables* on page 48
4. Modify DirX's root entry so that it contains two new attributes: `vendorName` and `vendorVersion`. Otherwise, the Administration server cannot connect to DirX.	*To add vendor attributes to DirX's root entry* on page 49
5. Add `groupOfUniqueNames` as an object class.	*To add the groupOfUniqueNames object class* on page 49
6. If you are going to be adding groups with the Policy Builder, change Siemens' rule that disallows the parent entry to use the `groupOfUniqueNames` object class and when it also contains a subentry with an `organizationalUnit` object class.	*To modify DirX's object class rule* on page 50

Table 13: DirX as a user location only

This step...	For details, see...
7. Rerun the `dirxadm` application with the following Select Access-specific file: `source bind.tcl`.	N/A.
8. Restart DirX.	DirX's documentation

To relocate Select Access's attribute file

1. Move the `dirxabbr-VN-VV.SelectAccess` file Select Access's default installation location (`<install_path>\schema\Siemens-DirX\`) to Siemens's `<install_path>\Client\Conf directory`.
2. Rename this file to `dirxabbr-ext.SelectAccess`.

To modify the global variables

1. Open `globalvar.tcl`. This file is one of the Select Access-specific files described in Table 12.
2. Define values for the variables it contains to customize the file for your organization. Table 14 explains the variables contained in this file.

Ensure the `country` and `organization` variables are identical to the ones in your existing DSA or you are not able to log to the database.

The `country` and `organization` variables combine to form a DN in the LDAP database.

Table 14: Global variables and their new values

For this variable...	Use this value...
`country`	Your country's Internet domain. For example, for Germany, type `DE`.
`organization`	Your organization's name. For example, `sample.com`.
`name`	The directory server's administrator name or user ID. For example, `JDoe`.
`defaultPW`	A default or customized password. This variable is case sensitive.
`saAdminName`	Select Access's security administrator's name or user ID. For example, `saAdmin`.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

Table 14: Global variables and their new values (Continued)

For this variable...	Use this value...
`saAdminPw`	A default or customized password. This variable is case sensitive. For example, `W3wizard`.
`ldapPort`	The default LDAP port number. For example, `389`. Change the port number if the default port is already being used by another application.

The login name, which is case insensitive, is made up of the `organization` and `name` variables.

To add vendor attributes to DirX's root entry

1. To build the directory server to include vendor attributes required by Select Access, run the dirxadm application with the corresponding script using the command outlined below:

   ```
   dirxadm <install_path>\schema\Siemens-DirX\
   add_vendor_attributes.adm
   ```

   This adds the `vendorName` and `vendorVersion` attributes to the `ldapRootDSE` object class.

The attribute values used for vendorName and vendorVersion are defined in GlobalVar.tcl.

2. Exit dirxadm.
3. To identify the directory as a Siemens server, run the dirxcp application with the corresponding script using the command outlined below:

   ```
   dirxcp <install_path>\schema\Siemens-DirX\
   extend_LDAP_Root.cp
   ```

   This script modifies the root DSA Entry to contain the attributes you added in step 1.
4. Exit `dirxcp`.

To add the groupOfUniqueNames object class

1. To add the `groupOfUniqueNames` object class, run the dirxadm application with the corresponding script using the command outlined below:

   ```
   dirxadm <install_path>\schema\Siemens-DirX\add_goun_nf.adm
   ```
2. Exit dirxadm.

To modify DirX's object class rule

1. To modify DirX's, object class rule, run the dirxcp application with the corresponding script using the command outlined below:

   ```
   dirxcp <install_path>\schema\Siemens-DirX\goun_rule.cp
   ```

   This script allows the parent entry of a `groupOfUniqueNames` object class to contain an entry with the `ogranizationalUnit` object class, which is a requirement of Select Access.

2. Exit dirxcp.

The CP Directory schema for v4.0

The Critical Path directory server v4.0 requires manual intervention before the schema uploads correctly. If this is your first time installing Select Access, Table 15 outlines the steps you must take to upload the CP Directory schema on Select Access.

Table 15: CP v4.0 overview

This step...	For details, see...
1. Stop all of your DSAs.	CP Directory's documentation
2. Copy Select Access-specific files to CP Directory's working folder. These files are needed to upload Select Access's schema data to your existing schema file.	*To upload Select Access schema and attribute data* on page 51
3. Use `odsadmin` to bind the manager to the DSA. This allows you to log into the directory as `manager`.	*To bind a manager to the DSA* on page 51

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

Table 15: CP v4.0 overview (Continued)

This step...	For details, see...
4. In CP's configuration file, change the `admin size limit` parameter to be something much higher than `50`. This limit controls the amount of data returned by CP in response to a search request. **Note:** HP recommends a value of at least `1000`. 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 administration size limit.	CP Directory's documentation
5. Start all of your DSAs.	CP Directory's documentation

To upload Select Access schema and attribute data

1. Locate the files you need. By default, all schema-related files are installed to `<install_path>/schema/Critical-Path/CP4.0/`.
2. Copy the following files to the`...\<SA_dsa_name>` CP Directory folder.
   - `schema-SA`: Contains the scripts that load the schema
   - `oidslocal-SA`: Is HP's schema attribute file
3. Append the contents of the `oidslocal-SA` file to your current `oidslocal` file.

To bind a manager to the DSA

1. Open a DOS window and change to the working directory, which is typically `ids\icon`.
2. Run the `odsadmin` application. This Critical Path-specific command starts the Select Access scripts.
3. To bind an administrator as manager, type the following command:

   ```
   bind -n N(Rcn(<manager_cn>)) -p <password>
   ```
4. Once bind has completed successfully, upload the Select Access-specific schema with the following command:

   ```
   @schema-SA~
   ```

5. Type a period (.) to exit `odsadmin` when the schema upload is complete.
6. To generate the Select Access-specific version of the `oidslocal` file for the DSA, run the `odsschema` application. This Critical Path-specific command rebuilds the LDAP schema.

⚠ This command could generate an error. If an error is reported, you can disregard it, as your system is not adversely affected by this command.

The CP Directory schema for v4.1

The Critical Path directory server v4.1 requires manual intervention before the schema uploads correctly. However, due to changes made to CP since v4.0, the steps required to achieve this integration are considerably streamlined. If this is your first time installing Select Access, Table 16 outlines the steps you must take to upload the CP Directory schema on Select Access.

Table 16: CP v4.1 overview

This step...	For details, see...
1. Create a new DSA.	Critical Path documentation
2. In CP's configuration file, change the `admin size limit` parameter to be something much higher than `50`. This limit controls the amount of data returned by CP in response to a search request. **Note:** HP recommends a value of at least `1000`. 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 administration size limit.	CP Directory's documentation
3. Upload the Select Access schema.	*To upload the Select Access schema* on page 52

To upload the Select Access schema

1. Locate the Select Access schema files. By default, the installer installs these files to the `<install_path>/schema/Critical-Path/CP4.1` directory.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

2. Upload the schema using the `ldapmodify` command. This command allows you to modify the default schema and import it over LDAP.

`ldapmodify` takes the name of the Select Access LDIF script as a parameter.

For explicit details on either of this command, see Critical Path's documentation.

3. Upload the LDIF schema files in the following order. If you do not upload these files in the order outlined below, unpredictable results can occur.
   - `ldapattribs.ldif`
   - `ldapocadd.ldif`
   - `ldapocupdate.ldif`
   - `ldapnb.ldif`

The Microsoft Active Directory schema

This directory server requires manual intervention before the schema uploads correctly. Table 12 outlines the steps you must consider before you upload Active Directory's schema on Select Access.

Microsoft Active Directory only allows passwords to be changed over SSL. Therefore, you cannot create accounts with passwords, modify passwords, or use password management unless SSL is enabled. For information on how to enable SSL on Active Directory so that it works with Select Access, see *To enable SSL with Active Directory* on page 54.

Table 17: Active Directory schema overview

This step...	**For details, see...**
1. Modify the registry.	*To modify the registry automatically* on page 54 OR *To modify the registry manually* on page 54
2. Activate the Select Access-specific schema.	*To activate the Active Directory schema* on page 55
3. Set schema permissions. **Note:** You must log in as Administrator to have the necessary write permissions.	*To set schema permissions* on page 55

To enable SSL with Active Directory

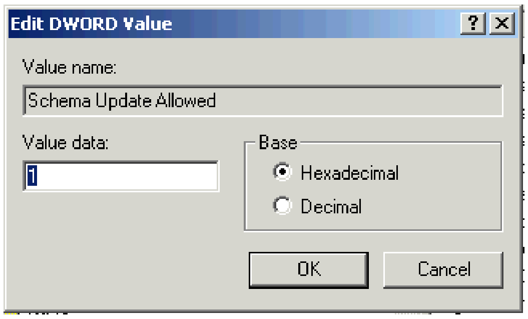
1. Run the Setup Tool for the Administration server. On the **Directory Server** setup screen, configure the connection information required to connect to your Active Directory server. However, do not check the **Use SSL** box.
2. To upgrade the schema, follow the procedure described in Table 17.
3. Run the Setup Tool for the Administration server again. On the **Directory Server** setup screen, check the **Use SSL** box.

To modify the registry automatically

1. Double-click the `adupdate.reg` registration file. By default, all schema related files are installed to `<install_path>\schema\ActiveDirectory\`.
2. Before the registry is modified, a confirmation message is displayed. Click **Yes** to confirm changes.
3. A message is displayed telling you whether the update has been successful.
   - If the update is successful, click **OK**.
   - If the update is not successful, manually modify the registry. For more information, see *To modify the registry manually*.

To modify the registry manually

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 just typed in step 4. As shown in Figure 5, the **Edit DWORD value** dialog appears.


Edit DWORD Value

Value name: Schema Update Allowed

Value data: 1

Base: (● Hexadecimal) Decimal

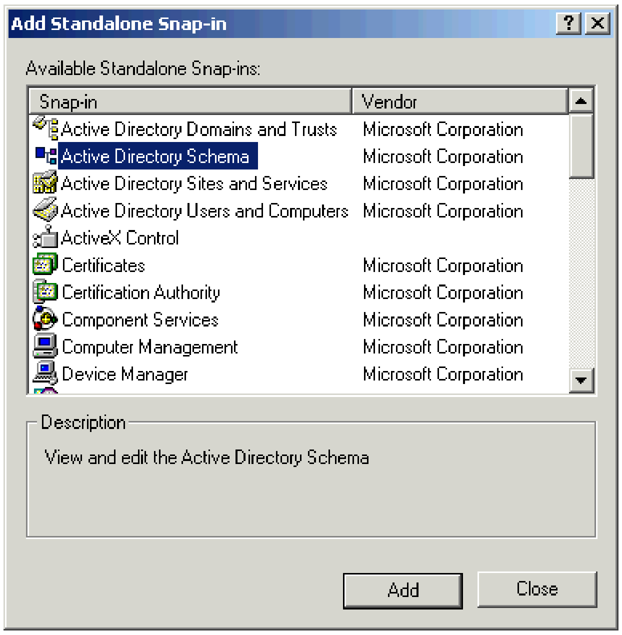
Figure 5: The Edit DWORD value dialog

6. In the **Value data field**, type `1` and click **OK**.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

To activate the Active Directory schema

1. Run the **Microsoft Management Console** application.
2. From the main menu, click **Console>Add/Remove snap-in**. The **Add/Remove snap-in** dialog appears.
3. Click **Add**. As shown in Figure 7, the **Add Standalone snap-in** dialog appears.



Snap-in	Vendor
Active Directory Domains and Trusts	Microsoft Corporation
Active Directory Schema	Microsoft Corporation
Active Directory Sites and Services	Microsoft Corporation
Active Directory Users and Computers	Microsoft Corporation
ActiveX Control	
Certificates	Microsoft Corporation
Certification Authority	Microsoft Corporation
Component Services	Microsoft Corporation
Computer Management	Microsoft Corporation
Device Manager	Microsoft Corporation

Description

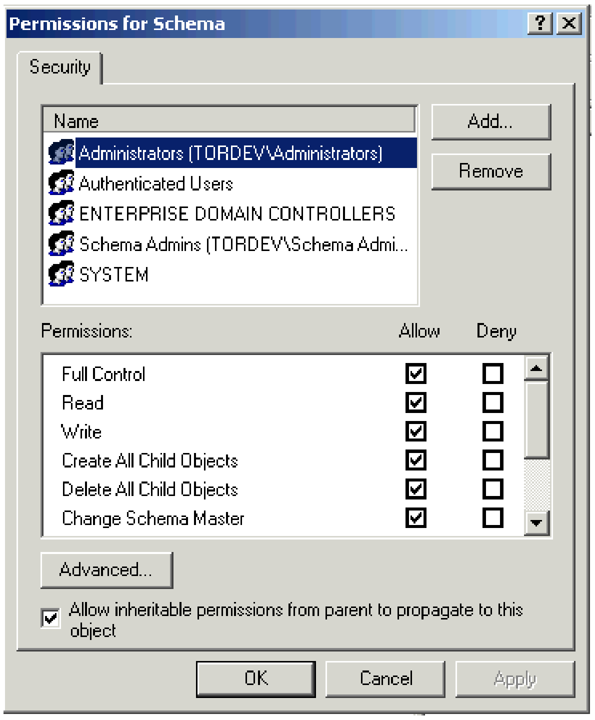
View and edit the Active Directory Schema

Figure 6: The Add Standalone snap-in dialog

4. Select **Active Directory Schema** and click **Add**. This activates the schema.

To set schema permissions

1. Run your schema by clicking **Start>Programs>Administrative Tools> Active Directory Schema.** This displays the **Console** window.
2. In the left pane, right-click **Active Directory Schema>Permissions.** As shown in Figure 8, the **Permissions for schema** dialog appears.


Permissions for Schema

Security

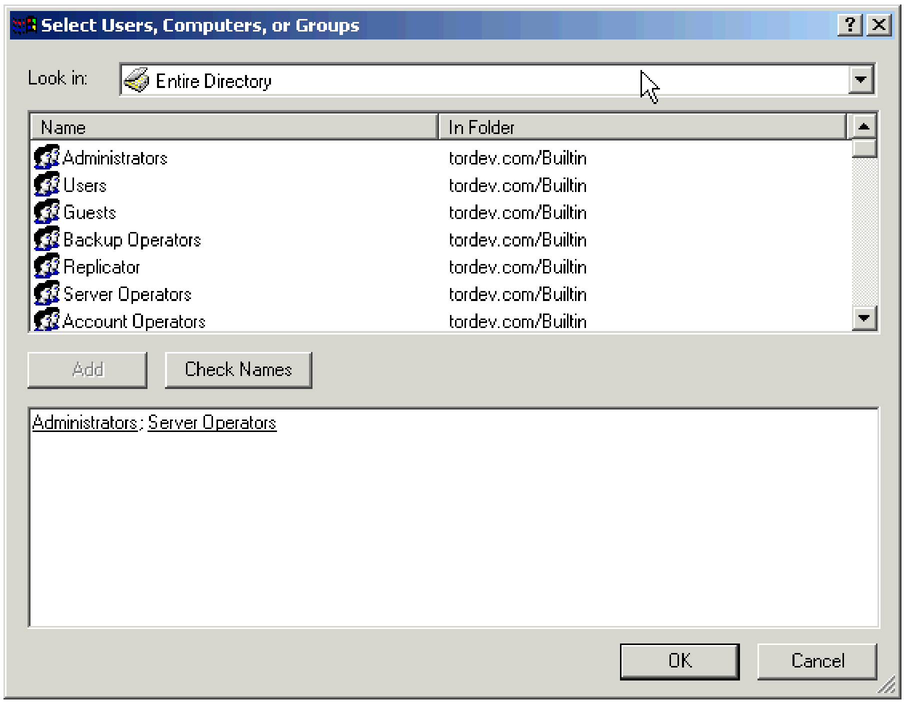
Name
Administrators (TORDEV\Administrators)
Authenticated Users
ENTERPRISE DOMAIN CONTROLLERS
Schema Admins (TORDEV\Schema Admi...
SYSTEM

Permissions:	Allow	Deny
Full Control	☑	☐
Read	☑	☐
Write	☑	☐
Create All Child Objects	☑	☐
Delete All Child Objects	☑	☐
Change Schema Master	☑	☐

☑ Allow inheritable permissions from parent to propagate to this object

Figure 7: The Permissions for schema dialog

3. Click **Add**. As shown in Figure 9, the **Select users, computers, or groups** dialog appears.


Select Users, Computers, or Groups

Look in: Entire Directory

Name	In Folder
Administrators	tordev.com/Builtin
Users	tordev.com/Builtin
Guests	tordev.com/Builtin
Backup Operators	tordev.com/Builtin
Replicator	tordev.com/Builtin
Server Operators	tordev.com/Builtin
Account Operators	tordev.com/Builtin

Administrators ; Server Operators

Figure 8: the Select users, computers, or groups dialog

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

4. Select the users, groups, or computers you want to assign full access rights to, and click **Add**. This moves the selected names to the bottom pane.
5. When you are finished, click **OK**.
6. In the **Permissions** section, enable **Full control** and click **OK**.
7. At the **Save console settings to Console** prompt, click **Yes**.
8. Enter a name in the **Save as** field and click **Save**.

The Oracle Internet Directory schema

This directory server requires manual intervention before the schema uploads correctly.

If this is your first time installing Select Access, Table 18 outlines the steps you must take to upload Oracle's schema on Select Access:

⚠ Ensure you perform each and every step as outlined in Table 18. Steps 4 and 5 are particularly critical to the successful integration of Select Access with OID. Otherwise, you cannot use the Policy Builder to connect to this directory server, nor can you browse or modify the Policy Store through a non-ssl port.

Table 18: Oracle overview

Step	For details, see...
1. Configure SSL settings to set an SSL Authentication type.	*To configure SSL settings* on page 58
2. Add a `namingContexts` attribute to the root DSE and change the default password encryption algorithm.	*To modify operational attributes* on page 58
3. If no entry exists, create one that corresponds to the same `namingContexts` attribute, and DN as outlined in step 1. By default no entry exists.	*To create an entry* on page 58
4. Rerun the Setup Tool and reconfigure the policy and/or user data locations on this directory server.	*To configure the Administration server* on page 60 in the *HP OpenView Select Access 6.0 Installation Guide*
5. Restart any directory server instances so changes are uploaded.	OID documentation

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

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** box at the top of the tab.
5. In the **SSL Authentication** field, select **SSL Server Authentication** from the list.
6. Click the **Apply** button.

To modify operational attributes

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`, `SHA`, and `UNIX crypt algorithm`. Select Access supports all algorithm options in the drop-down list except for MD4.
5. Click the **Apply** button.

To create an entry

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 operational attributes* on page 58.
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**.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

7. In the **Mandatory Properties** section, type in the DN (for example, `mycompany.com`).
8. Click **OK**.

The eTrust Directory server schema

Integrating Select Access with eTrust Directory on Windows 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 LDAP 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 the fax and telephone number fields on the `registration_form.html` form that is located on the server are mandatory, we recommend you modify the form by placing an asterisk beside the fields.

If this is your first time installing Select Access, Table 19 outlines the steps you must take to upload eTrust's schema on Select Access.

Table 19: eTrust overview

Step...	**For details, see...**
1. Copy the eTrust schema script to an eTrust location.	*To copy the eTrust schema script* on page 60
2. Modify the directory server schema to make `uniqueMember` an optional attribute.	*To modify the directory server schema* on page 60
3. Modify the group file for the DSA that you want Select Access to reference.	*To modify the DSA initialization file to change the group schema file* on page 61

Table 19: eTrust overview (Continued)

Step...	**For details, see...**
4. Increase the `max-op-size` parameter from its default value. This limit controls the amount of data returned by eTrust in response to a search request. eTrust's default value for this parameter is too low for the needs of Select Access. **Note:** HP recommends a value of at least `1000`. 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.	*To modify the maximum operation parameter* on page 62
5. Start the DSA to load the modified group file.	*To load the modified DSA group file* on page 62

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 modify the directory server schema

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 `must-contain` and then add it to `may-contain`, as shown in Code example 3. This makes the schema compatible with Select Access's object classes and their corresponding attributes.
5. Save the modified `x500.dxc` file.

Code example 3: Modified x500.dxc file

```
schema set object-class standardObjectClass:17 = {
        name = groupOfUniqueNames
        subclass-of top
        must-contain
```

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

```
            commonName
        may-contain
            uniqueMember,
            description,
        organizationName,
            organizationalUnitName,
            owner,
            seeAlso,
            businessCategory
};
```

To modify the DSA initialization file to change the group schema file

1. Open your group 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.
2. Add the following line to the end of the group file, as shown in Code example 5:
   `source "SAeTrustSchema.dxc";`

   This adds the Select Access-specific eTrust schema script to the group file.
3. Open the initialization file for the DSA that you want Select Access to reference. Initialization files are located in the following location:

   `<eTrust_install_path>/dxserver/config/servers`.

   Initialization files have a `.dxi` extension. eTrust ships a sample initialization file, called `democorp.dxi`, shown in Code example 4.
4. Determine the schema group file by looking in the schema section of the initialization file, which begins with the following line: `# schema`. eTrust Directory group files have the following extension: `.dxg`.

   In the sample initialization file in Code example 4, the group file is as follows:
   `source "../schema/default.dxg";`
5. Save the modified `.dxg` file.

Code example 4: Sample democorp.dxi initialization file

```
# logging and tracing
close summary-log;
close trace-log;
source "../logging/default.dxc";

# LEGACY licence
# source "../licence/default.dxc";

# schema
clear schema;
```

```
source "../schema/default.dxg";

# access controls
clear access;
# source "../access/";

# knowledge
clear dsas;
source "../knowledge/sample.dxg";

# operational settings
source "../settings/default.dxc";

# service limits
source "../limits/default.dxc";

# database
source "../database/democorp.dxc";

# replication agreements (rarely used)
# source "../replication/";
```

Code example 5: Sample default.dxg group file

```
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";
```

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 = 1000`
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`.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

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 Users Tree gets populated.

Updating certain directory servers' schemas

Because Select Access requires a tight integration with directory servers, schemas for some require 6.0-specific updates. This ensures that policy and user 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. Schemas for the following directory servers schemas must be:

- Siemens DirX. For more information, see *To upgrade Select Access schema files for Siemens DirX* on page 63.
- eTrust. For more information, see *To upgrade Select Access schema files for eTrust* on page 64.
- CP Directory 4.0 and 4.1. For more information, see one of the following sections:
  - *To upgrade Select Access schema files for CP Directory v4.0* on page 64
  - *To upgrade Select Access schema files for CP Directory v4.1* on page 66

If your directory server(s) require a schema update, you must do this before you install and run Select Access.

To upgrade Select Access schema files for Siemens DirX

1. Stop the Siemens DirX directory server.
2. Copy files from `<SelectAccess_install_path>/schema/Siemens-DirX/` to the corresponding DirX folder. For details on what these files do, see *The Siemens DirX schema for policy and user data* on page 40. Table 20 summarizes which files you need to copy to a new DirX disk location.

Table 20: Files to copy for Siemens upgrade

DirX disk location	Select Access files copied
`<DirX_install_path>\Client\Conf`	• `dirxabbr-ext.SelectAccess` **Note:** If you want to save the previous version of this file, rename it by prepending a word to it to distinguish it from the file used for upgrading. For example, `old_dirxabbr-ext.SelectAccess`.

Table 20: Files to copy for Siemens upgrade (Continued)

DirX disk location	Select Access files copied
`<DirX_install_path>\scripts\security\SelectAccess`	Depending on which version of Select Access you are upgrading from, different files must be copied: • If upgrading from Select Access 5.0 to 6.0: — `upgrade-5.0_to_6.0.bat` — `upgrade_schema_ext_for_SelectAccess_5.0_to_6.0.adm` • If upgrading from Select Access5.1 or 5.2 to 6.0: — `upgrade-5.1_to_6.0.bat` — `upgrade_schema_ext_for_SelectAccess_5.1_to_6.0.adm` In addition, the following files must be copied: • `upgrade_subshema_ext_for_SelectAccess_5.x_to_6.0.cpx` • `Extend_LDAP_Root.cp` • `GlobalVar.tcl`

3. Start the directory server.
4. Run `upgrade.bat`.If you need to check anything after the upgrade, this file outputs to `upgrade.log`.

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. 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 Users Tree gets populated.

To upgrade Select Access schema files for CP Directory v4.0

1. Relocate Select Access files to the corresponding CP Directory location:
   — Locate the files you need. By default, all schema-related files are installed to `<install_path>/schema/Critical-Path/CP4.0`.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

— Copy the files outlined in Table 21 to the CP Directory folder, which is typically `ids\icon\<SA_dsa_name>`.

Table 21: Files to upgrade schema for CP Directory v4.0

For this version of Select Access...	Copy these files...
5.0	• `upgradebaltschema_50_to_60`: Contains the upgraded scripts that load the schema when upgrading from 5.0 to 6.0. • `oidslocal`: Configures the directory before loading the schema.
5.1 or 5.2	• `upgradebaltschema_51_to_60`: Contains the upgraded scripts that load the schema when upgrading from versions 5.1 or 5.2 to version 6.0. • `oidslocal`: Configures the directory before loading the schema.

2. Bind a manager to the DSA:
   - From a command prompt, change to the working directory, which is typically `ids\icon\<SA_dsa_name>`.
   - Run the `odsadmin` application. This Critical Path-specific command starts the Select Access scripts.
   - To bind an administrator as manager, type the following command:

     ```
     bind -n N(Rcn(<admin CN>)) -p <password>
     ```
   - Once `bind` has completed successfully, upload the Select Access-specific schema with one of the following commands, depending on what version you are upgrading to:

     If you are upgrading from Select Access 5.0 to 6.0, enter the following command:

     ```
     @upgradebaltschema_50_to_6.0~
     ```

     If you are upgrading from Select Access 5.1 or 5.2 to 6.0, enter the following command:

     ```
     @upgradebaltschema_51_to_6.0~
     ```
   - To generate the Select Access-specific version of the `oidslocal` file for the DSA, run the `odsschema` application. This Critical Path-specific command rebuilds the LDAP schema.

⚠ This command can generate an error. If an error is reported, disregard it. Your system is not adversely affected by this command.

3. Start all of your DSAs. For details see CP Directory's documentation.

To upgrade Select Access schema files for CP Directory v4.1

1. Locate the files you need. By default, all schema-related files are installed to `<install_path>/schema/Critical-Path/CP4.1`.
2. Depending on the version of Select Access you are upgrading from, upload the corresponding file:
   - From Select Access 5.0: Upload the file `4.1_upgrade_schema_ext_for_SelectAccess_5.0_to_6.0.ldif`.
   - From Select Access 5.1 or 5.2: Upload the file `4.1_upgrade_schema_ext_for_SelectAccess_5.0_to_5.1.ldif`.

   For more information on how you can upload either LDIF file, see *To upload the Select Access schema* on page 52.

Tuning directory servers by indexing 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 uploaded your directory server's schema successfully.

The key to proper indexing is knowing what type of searches are most likely to occur. The Policy Validator frequently evaluates users with the attributes listed below. Therefore, if you want to increase Select Access's performance, you must index these attributes. By indexing these attributes, you create a cache large enough to prevent the Policy Validator from repeated (and unnecessary) disk access:

It is always good practice to index attributes before you place Select Access and your directory server into full-scale production.

- `objectClass`
- `samAccountName`
- `userPrincipalName`
- `cn`
- `uid`
- `ou`
- `memberUid`
- `uniqueMember`
- `member`
- `nxRole`
- `nxUrl`
- `nxSelectIdRule`
- `nxUnknownUserRule`
- `nxRule`
- `nxManagement`
- `nxResource`
- and any attributes you have used in role definitions

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

For information on how to index attributes on your directory server, see your server's documentation.

Setting up replication and referrals

Depending on your preexisting directory system topology, there can be any number of ways you have chosen to structure your user data. The more entries 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 user 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.

For replication and referrals to work on a Select Access-protected system, the login for all servers must be the same.

Multiple servers and user locations

Because Select Access supports multiple user 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 9, 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 user location with its own set of user entries, groups, and roles.

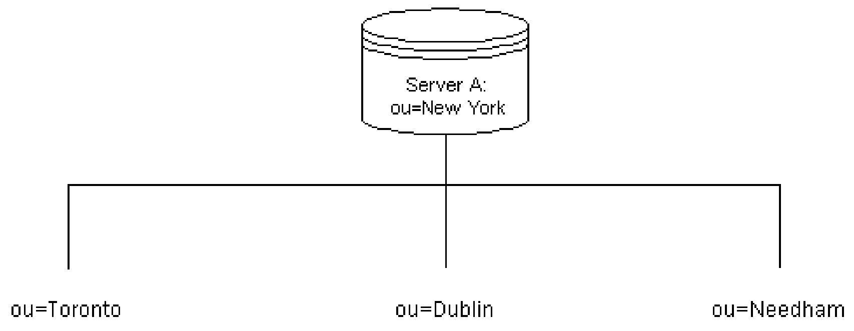


Figure 9: Example directory tree with multiple user locations

However, rather than storing user data as branches on this tree, you can also move the data on these branches into separate servers, as shown in Figure 10.

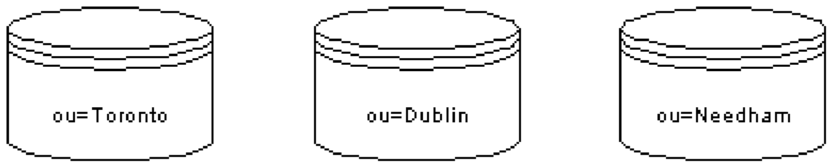


Figure 10: Directory branches separated into different servers

This configuration still keeps user locations unique, but gives you the added benefit of being able to separate the user 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 11.

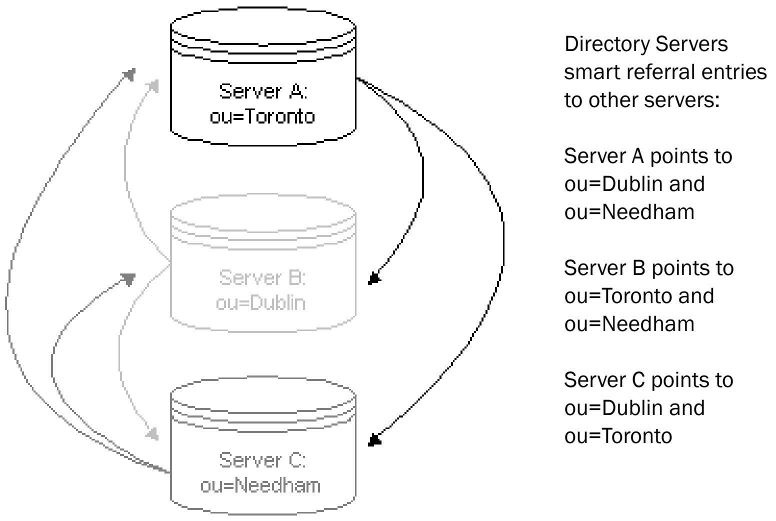

Server A:
ou=Toronto
Server B:
ou=Dublin
Server C:
ou=Needham
Directory Servers smart referral entries to other servers:
Server A points to ou=Dublin and ou=Needham
Server B points to ou=Toronto and ou=Needham
Server C points to ou=Dublin and ou=Toronto

Figure 11: Referrals: multiple directory servers with multiple user locations

To design your topology for replication, you might take these servers and create replicas of them on different hosts, as shown in Figure 12. These replicas must be readable and writable (that is, a multi-master environment).

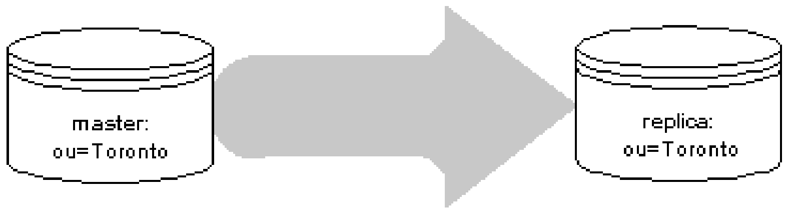


Figure 12: Replication: duplicated multiple directory servers

Using multiple-master replication with Select Access

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

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

supports multiple-master replication because it minimizes downtime if a single server fails.

⚠ 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 entry as needed. The Policy Validator must be able to write to the directory server in question.

⚠ In order for replication to work with Active Directory, double-click the `regupdate` file on any replicated machine running Active Directory. Run `regupdate` from the following location: `<install_path>\schema\ActiveDirectory\`.

If you do not run `regupdate` on all your replicated servers, you cannot log into any of your directory servers, including the master, using the Policy Builder.

Tips for using replicated directory servers with Select Access

To ensure that replicated directory servers work seamlessly with Select Access clients, take care to do the following:

- 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 cannot spend a lot of time trying to connect to a machine that is unreachable. If an aggressive connection timeout is not configurable 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.
- Ensure the schemas on replicated servers are the same (attributes and version numbers), particularly when you have modified the user schema to upload with Select Access. Otherwise, replication operations fail. We recommend that you maintain the schema on the master server only and replicate the same schema to other servers.
- All replicated copies of Policy Store data need to be writable. For areas of the Policy Store that are updated by the Policy Validator, you need to further ensure that directory servers redistribute data

to replicas as soon as information changes. This ensures data consistency among all replicas.

⚠ Inconsistent data can cause the Policy Validator to deny access to a user who would normally be allowed to access a given resource.

- If you are using user accounts that allow users to self-manage passwords or profile data, then replicated copies of user data need to be writable as well.

Using replication and referrals with Select Access

When user data is distributed over several servers, you need to define these relationships among your directory servers so Select Access components are pointed at the correct directory for the data that they request:

- Policy Builder: This component needs to find the correct directory server, particularly when multiple user locations have been configured across multiple directory servers. If a branch of a user tree is on a different server than the one that has been defined as the root, Policy Builder needs to know where to find the user data so it can be represented in the Policy Matrix.
- Policy Validator: When evaluating whether to allow or deny access to a resource, the Policy Validator needs to know where on the network to perform the user lookup. To facilitate this, the Policy Validator uses an optional bootstrap parameter called `ldapUseReferral`. This parameter specifies whether or not the directory server generates referrals. For details, see Appendix B, *Editing the bootstrap configuration files*, in the *HP OpenView Select Access 6.0 Installation Guide*.
- Administration server: This component needs to know which directory servers are replicated. You configure which directory servers are replicated when you do a **Custom** setup and use the **Replicated Directory Servers** setup screen. For details, see Chapter 5, *Configuring the Administration server*, in the *HP OpenView Select Access 6.0 Installation Guide*.

Smart referral

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.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

Directory entries acting as a smart referral are defined with:

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

LDAP URL syntax

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 set by Figure 11, a search referral to Server B might look like this:

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

How to set up multiple user locations

To prevent user authentication ambiguities from occurring when the Policy Validator caches user entries, Select Access does not allow you to create user locations that have overlapping root DNs. Overlapping root DNs occur when you try using:

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

Ensure that the root DNs of all your user locations have a unique name.

Sample scenario: multiple directory servers

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

  `o=mycompany.com, ou=americas_users`

  `o=mycompany.com, ou=europe_users`

  `o=mycompany.com, ou=asia_users`

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

Chapter 6

Setting up your Web server

A Web server hosts Web pages, applications, or other content, and is protected by an Enforcer plugin. In a typical network configuration, the Web server sits outside of your firewall. The Enforcer plugin protects the content that the Web server manages by enforcing the decisions the Policy Validator makes. Therefore, the Enforcer plugin can do little to protect resources on a Web server without one or more access policies. The access policy for resources also requires a method of authenticating the user who requests access to it.

When to manually modify a Web server's configuration file

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. For details, see *To run the installer in Console mode on a clean host machine* on page 47 in the *HP OpenView Select Access 6.0 Installation Guide*.

In these cases, you need to know how to manually make the required modifications. Table 22 provides an overview on how to prepare a Web server for use with Select Access.

Table 22: Preparing your Web server

Step...	**For details, see...**
1. Modify your server's configuration file(s) to load the Enforcer plugin upon start up.	*Preparing your server to use the Enforcer plugin* on page 74
2. Determine what you need to do to your Web server and its resources to get content protected.	*Other steps to take to secure your resources* on page 90
3. If your Web server performs virtual hosting, configure the corresponding Enforcer plugin accordingly.	*Configuring your Enforcer plugin for virtual hosting* on page 90
4. If your Web server requires a custom plugin to include site-specific data, create one for your server.	*Using custom Web server plugins to include site-specific data* on page 93
5. If you plan to perform form-based login, you may want to customize default form templates shipped with Select Access.	*Select Access forms used by Enforcer-protected Web servers* on page 93

Preparing your server to use the Enforcer plugin

Before you install and configure an Enforcer plugin on your Web server, start by verifying that the Web server is installed correctly by accessing the main page over HTTP. Once you are sure that the Web server is functional, you can install the corresponding Enforcer plugin for it. Table 23 outlines the way in which Web servers can be Enforcer-protected.

Due to the dissolution of the relationship between Netscape and Sun, iPlanet Web servers have been renamed to Sun ONE. Note that this name change affects the name of the Enforcer plugin for this server. The Sun ONE (iPlanet) Enforcer plugin still supports existing iPlanet Web servers as well as the newly named Sun ONE v6.0 Web servers.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

Table 23: Overview of Enforcer-protected Web servers

This service...	For configuration details, see...
IIS Web server	*Configuring the IIS Web server* on page 75
iPlanet or Sun ONE Web servers	*Manually configuring the iPlanet 4.0 Web server* on page 83 OR *Manually configuring the Sun ONE v6.0 Web server* on page 86
Apache Web server	*Configuring the Apache Web server* on page 88

Configuring the IIS Web server

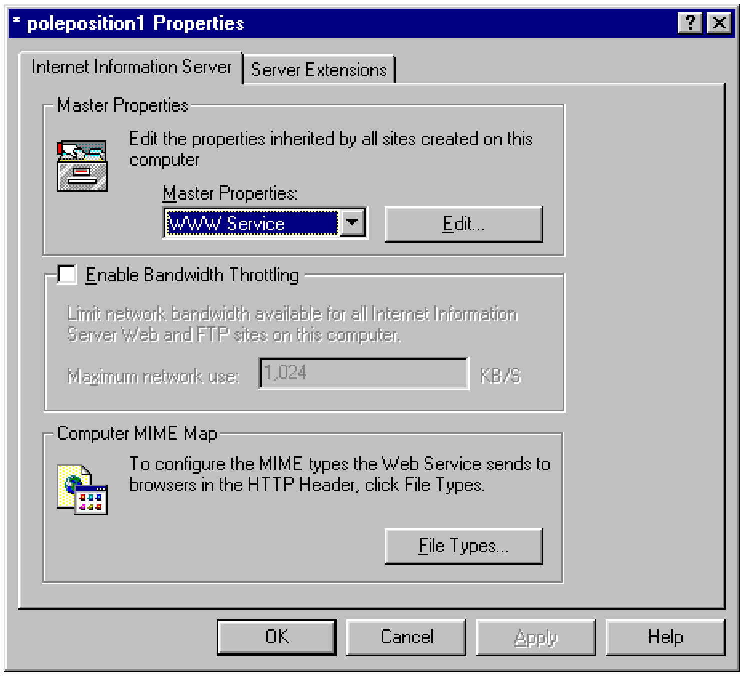
Typically, the Setup Tool configures your Web server to load the corresponding Enforcer plugin for it. However, if you need to manually configure your server to load the Enforcer plugin, follow the three steps outlined in Table 24.

Table 24: Configuring the IIS Web server overview

Step...	For details, see...
1. Define a valid IP address for this server.	*To assign a valid IP address to the Web server* on page 75
2. Install the `IISPlugin32.dll` filter.	*To install the filter for the IIS Enforcer plugin* on page 77
3. Follow additional configuration tips as needed.	*Configuration tips for your server and the IIS Enforcer plugin* on page 80

To assign a valid IP address to the Web server

1. Start Microsoft Management Console:
   a. Click **Start>Run**. The **Run** dialog appears.
   b. Type `mmc` in the `Open` field and click **OK**. The **Microsoft Management Console** window appears.
2. Under **Internet Information Server**, right-click the Web site to which you want to assign an IP address and choose **Properties**.
3. On the **Internet Information Server** tab, in the **Master Properties** list, select **WWW Service**, as shown in Figure 13.


* poleposition1 Properties

Internet Information Server | Server Extensions

Master Properties

Edit the properties inherited by all sites created on this computer

Master Properties: WWW Service

Edit...

☐ Enable Bandwidth Throttling

Limit network bandwidth available for all Internet Information Server Web and FTP sites on this computer.

Maximum network use: 1,024 KB/S

Computer MIME Map

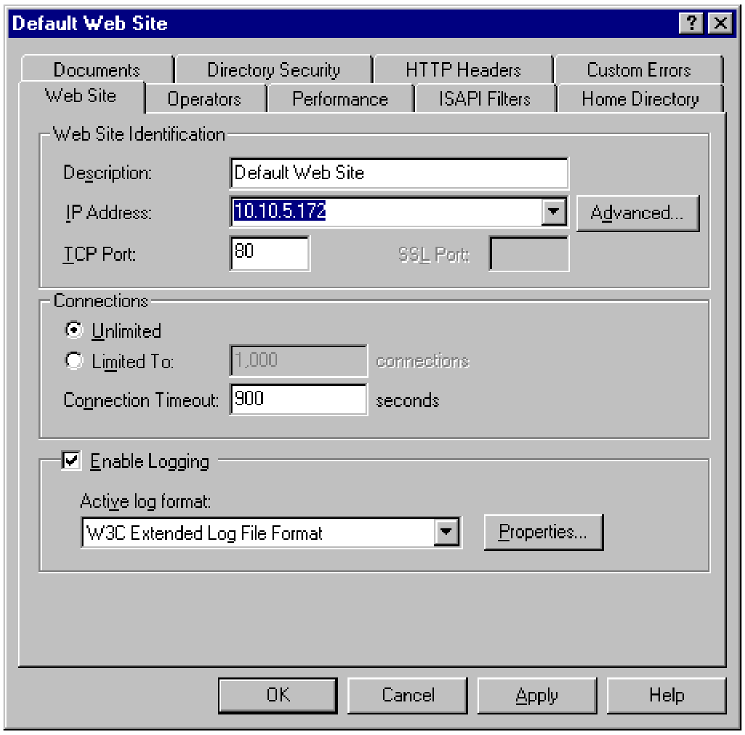
To configure the MIME types the Web Service sends to browsers in the HTTP Header, click File Types.

File Types...

OK | Cancel | Apply | Help

Figure 13: The Internet Information Server tab

4. Click the **Edit** button. The **WWW Service Master Properties** dialog appears.
5. Select the **Web Site** tab, as shown in Figure 14.


Default Web Site

Documents | Directory Security | HTTP Headers | Custom Errors
Web Site | Operators | Performance | ISAPI Filters | Home Directory

Web Site Identification

Description: Default Web Site

IP Address: 10.10.5.172

Advanced...

TCP Port: 80

SSL Port:

Connections

● Unlimited

○ Limited To: 1,000 connections

Connection Timeout: 900 seconds

☑ Enable Logging

Active log format: W3C Extended Log File Format

Properties...

OK | Cancel | Apply | Help

Figure 14: The Web Site Properties dialog

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

6. Enter a valid IP address in the **IP Address** field.
7. Click **OK**.

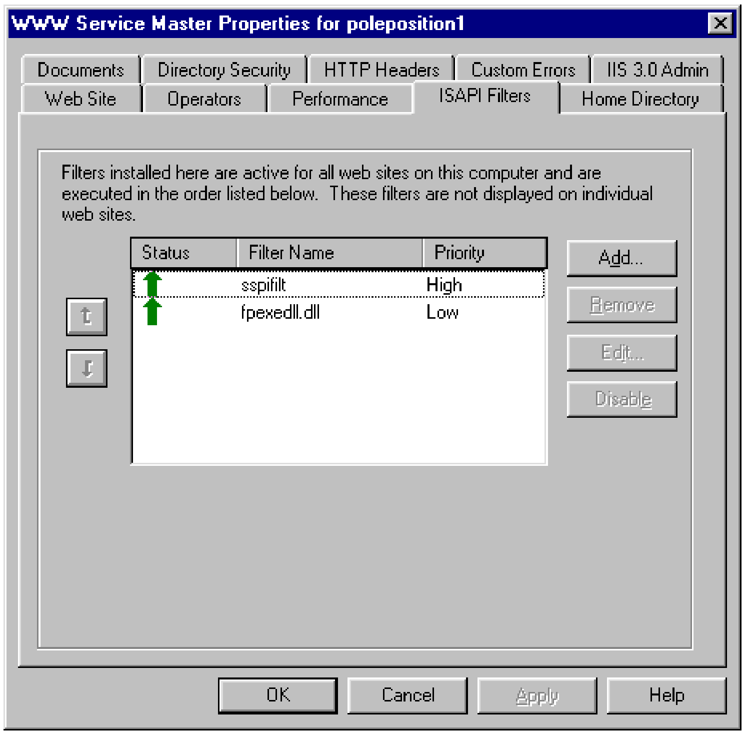
To install the filter for the IIS Enforcer plugin

1. Start Policy Validator.
2. Ensure that you set up the Enforcer plugin to query the correct Policy Validator.

⚠ The IIS filter does not install correctly unless the Policy Validator is running and the specific instance of the IIS Enforcer plugin is configured. To configure the IIS Enforcer plugin, see *Configuring the Enforcer plugin* on page 139 of the *HP OpenView Select Access 6.0 Installation Guide*.

---

3. From the **Microsoft Management Console** window, under **Internet Information Server**, right-click the host name and choose **Properties**.
4. As shown in Figure 13, on the **Internet Information Server** tab, in the **Master Properties** list, select **WWW Service**.
5. Click the **Edit** button. As shown in Figure 15, the **WWW Service Master Properties** dialog appears.

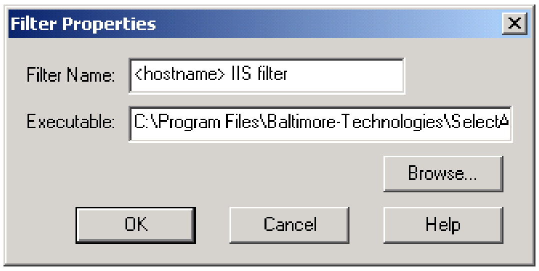

WWW Service Master Properties for poleposition1

Filters installed here are active for all web sites on this computer and are executed in the order listed below. These filters are not displayed on individual web sites.

Status	Filter Name	Priority
⬆	sspifilt	High
⬆	fpexedll.dll	Low

Figure 15: The WWW Service Master Properties dialog

6. On the **ISAPI Filters** tab, click the **Add** button. As shown in Figure 16, the **Filter Properties** dialog appears.


Filter Properties

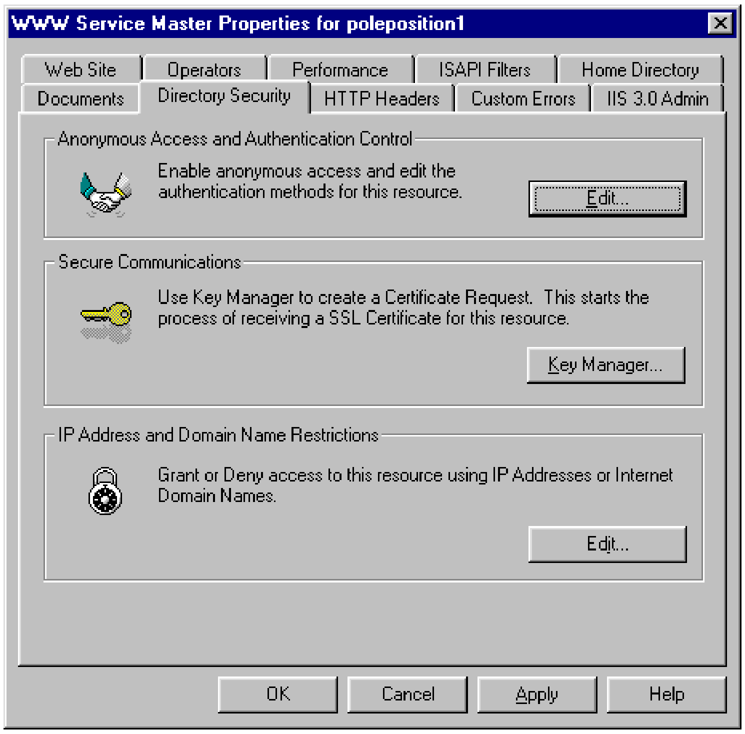
Filter Name: <hostname> IIS filter

Executable: C:\Program Files\Baltimore-Technologies\SelectA

Figure 16: The Filter Properties dialog

7. In the **Filter Name** box, enter a descriptive name for the filter (for example, `Select Access IIS Filter`).
8. Click **Browse** and locate `IISPlugin32.dll`. The path appears in the **Executable** box.

   By default, this file is installed in `<install_path>\bin\`.
9. Click **OK**, then click **OK** on the **WWW Service Master Properties** dialog.
10. Select the **Directory Security** tab. This tab is shown in Figure 17.


WWW Service Master Properties for poleposition1

Anonymous Access and Authentication Control

Enable anonymous access and edit the authentication methods for this resource.

Secure Communications

Use Key Manager to create a Certificate Request. This starts the process of receiving a SSL Certificate for this resource.

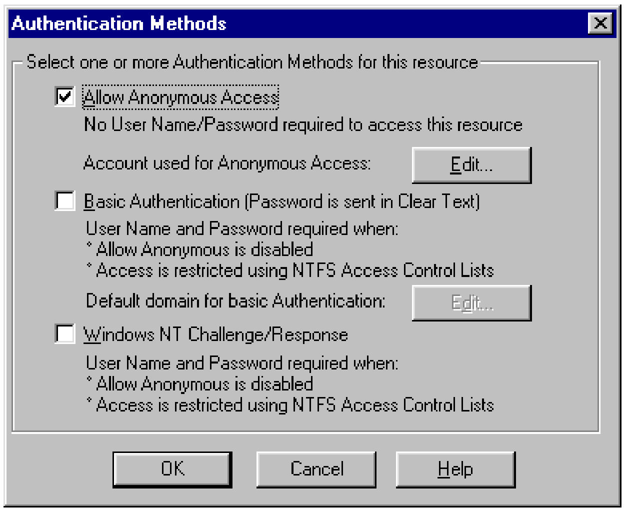
IP Address and Domain Name Restrictions

Grant or Deny access to this resource using IP Addresses or Internet Domain Names.

Figure 17: The Directory Security tab

11. Under **Anonymous Access and Authentication Control**, click the **Edit** button.
12. As shown in Figure 18, in the **Authentication Methods** dialog, make sure only the **Allow Anonymous Access** option is selected (in Windows 2000 the option is called **Anonymous Access**), then click **OK**.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.


Authentication Methods

Select one or more Authentication Methods for this resource

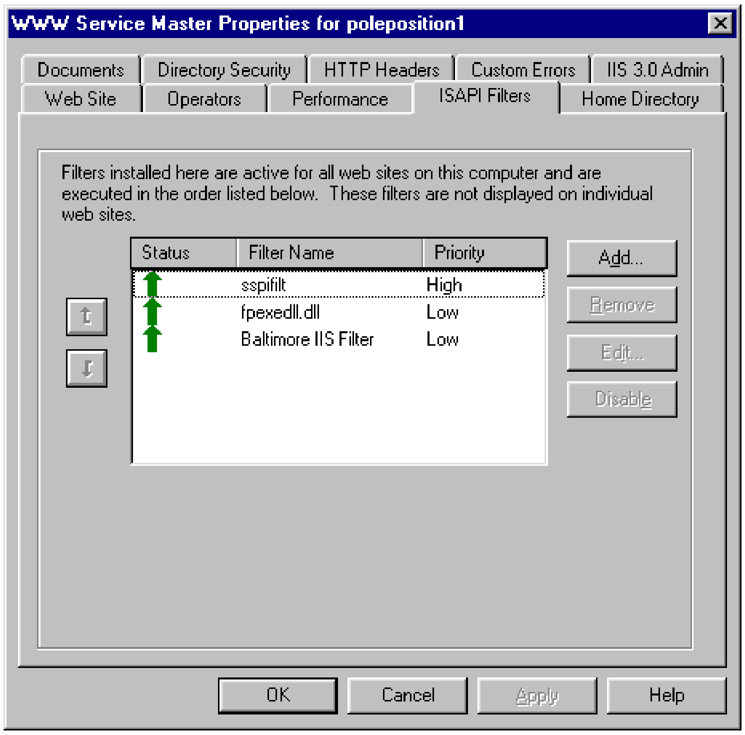
[✓] Allow Anonymous Access
No User Name/Password required to access this resource
Account used for Anonymous Access: Edit...

[ ] Basic Authentication (Password is sent in Clear Text)
User Name and Password required when:
* Allow Anonymous is disabled
* Access is restricted using NTFS Access Control Lists
Default domain for basic Authentication: Edit...

[ ] Windows NT Challenge/Response
User Name and Password required when:
* Allow Anonymous is disabled
* Access is restricted using NTFS Access Control Lists

Figure 18: The Authentication Methods dialog

13. Click **OK** to close the **Properties** dialog.
14. Stop and restart the Web server via **Control Panel>Services**.
15. Under **Internet Information Server**, right-click the host name and choose **Properties**.
16. Select the **ISAPI Filters** tab. If the IIS filter has loaded correctly, a green arrow is shown in the **Status** column beside the filter. If the filter has not loaded correctly, a red arrow (or no arrow) is shown, as demonstrated by Figure 19.


WWW Service Master Properties for poleposition1

Filters installed here are active for all web sites on this computer and are executed in the order listed below. These filters are not displayed on individual web sites.

Status	Filter Name	Priority
⬆	sspifilt	High
⬆	fpexedll.dll	Low
⬆	Baltimore IIS Filter	Low

Figure 19: The ISAPI Filters tab

17. If the IIS filter has not loaded correctly, do the following:
    - On the **ISAPI Filters** tab, select the IIS filter and click **Remove**, then click **OK**.
    - Stop and restart the Web server.
    - Reinstall the IIS filter.

Configuration tips for your server and the IIS Enforcer plugin

The IIS Enforcer plugin intercepts all Web server requests. As a result, the IIS Enforcer plugin checks the authorization for all the linked contents from that page. Set up Select Access to limit authentication to only main content pages for the following reasons:

- To get faster results because Policy Validator is not being called to authenticate users against each file
- To prevent multiple updates to the access control HTTP cookie. For details, see the section *Forcing user logout* on page 129
- To make maintenance easier because the site is divided into access controlled and non-access controlled areas

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

Table 25 outlines how to authenticate only main content pages.

Table 25: Configuration tips

If you have...	Then, do this...
One server	1. Organize access controlled and non-access controlled files into separate directory trees. 2. For the non-access controlled directory tree, disable SelectID and insert an allow rule in the Unknown user column.
Two servers	1. Set up a separate Web server from the one running the Select Access plugin. 2. Place access controlled content on the server with the Select Access plugin. 3. Place non-access controlled content on the other server.

If you are configuring the IIS Enforcer plugin and are also installing the WSE Enforcer plugin to protect web services, the IIS Enforcer plugin must be configured to ignore HTTP SOAP requests, or it will incorrectly attempt to validate the request using the information in the HTTP headers.

To allow SOAP requests to bypass IIS Enforcer plugin security, add the web services' relative URLs to the **Ignored Filenames** list. For more information, see *To create a list of ignored filenames* on page 157 of the *HP OpenView Select Access 6.0 Installation Guide*.

To configure your IIS and Select Access to use IIS's automatic logon mechanism

1. Configure the IIS Web server to perform user authentication with Integrated Windows authentication. For details, see IIS's documentation.
2. Based on your network topology, determine ho the authentication transfer between the browser and the Web server needs to be performed. You can choose either NTLM or Kerberos. You must configure an Select Access authentication server for either of these methods with the Policy Builder.

This method allows IIS—rather than the Policy Validator—to automatically authenticate users that have logged onto a desktop using NTLM or Kerberos.

⚠ Deploying an Integrated Windows authentication solution requires that you configure both your IIS Web server and Select Access to support this mechanism. As a result, both Microsoft and HP have specific requirements that you must meet to ensure that Integrated Windows authentication is successfully implemented. For details, see Table 26.

Table 26: Integrated Windows Authentication requirements list

This requirement...	Specific details on it...
HP-driven requirements	
Server-side limitations:	
Software support	• *Web server*: Must be IIS4 on Windows NT or IIS5 on either Windows NT or Windows 2000
	• *Controller/authentication type combination*: Must be either: — Windows 2000 Domain Controller with NTLM or Kerberos — Windows NT Domain Controller with NTLM only **Note:** You cannot use an IP address or a fully qualified domain name if you want to use Kerberos on a Windows 2000 Domain Controller. Instead, you must resolve an address or the DN to a Service Principal Name (SPN) of the service or of the name of the account used to run the service. If the address cannot be resolved, Kerberos fails. To do this, ensure you register this information on your directory server so addresses or DNs can be resolved. For details, see Microsoft's documentation.
Select Access deployment:	
Policy Validator	• *Operating systems*: Policy Validators that validate users with an Integrated Windows authentication server plugin can be installed on any platform.
	• *Domains*: Policy Validator can be running on any platform.
Microsoft-driven requirements	
Server-side limitations:	
IIS configuration	• *Properties:* Enable Integrated Windows Authentication (IWA). Otherwise, desktop authentication will not occur.
	• *Domains:* The server must either share the same Windows-based domain as the client or the server must be one of the trusted Windows-based domains of the client machine. Otherwise, permissions to resources cannot be granted.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

Table 26: Integrated Windows Authentication requirements list (Continued)

This requirement...	Specific details on it...
Permissions setup	• *NTFS privileges*: Ensure the user has the corresponding NTFS permissions for the resources in question. That is, set one of the following Types of Access: — **No Access** — **Read** — **Change** — **Full Control** **Note:** Ensure access required by a Web page is applied consistently. For example, if you assign **Full Control** to `Welcome.htm`, but **No Access** to `Logo.gif`, the user is not able to see the graphic.
Client-side limitations:	
Software support	• *Operating systems*: Can be one of Windows NT, Windows 2000, or Windows XP
	• *Browsers*: Must be IE v5.x or higher
IE configuration	Basic requirement: • *Domains:* Client must either share the same Windows-based domain as the server, or the server's domain must be one of the client's trusted Windows-based domains. Otherwise, permissions to resources cannot be granted.
	Requirements for automatic logon: • *IE v5.x properties:* Ensure you have not changed the default **Intranet zone** security setting from **Automatic logon**. • *IE v6.x properties:* Ensure you check the **Enable Integrated Windows Authentication** box in the **Advanced** tab of the **Internet Options** dialog. This box is not checked by default. • *URL evaluation:* Credentials are only passed when the URL is evaluated to be a local/intranet site. If the URL contains any periods (.), the URL is always interpreted as being an Internet site. For example, both 10.0.0.1 and `www.mycompany.com` are examples of URLs that are not passed. However, `http://mycompany` is an example of a URL that is passed. **Tip**: Addresses included in the **Intranet zone** in IE are exempted from this limitation. For details on how to add a URL to a specific security zone, see IE's online help.

Manually configuring the iPlanet 4.0 Web server

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 ONE (iPlanet)

Enforcer plugin (also used by iPlanet), you need to edit your `obj.conf` file to load the enforcer module and several functions.

---

iPlanet 4.0 Web server cannot handle virtual domains.

To configure your server on Windows

1. Open `obj.conf`.

   By default, this file is located in:

   ```
   <installation root> netscape\server4\https-
   <server name>.com\config\
   ```

   where:

   — *`<installation root>`* is the installation location of your server

   — *`<server name>`* is the name of your server.

2. Locate the following line in `obj.conf`:

   ```
   Init fn=load-types mime-types=mime.types
   ```

3. Add the following line after it:

   ```
   Init fn="load-modules"    shlib=
   "<install_path>/iplanet_web32.dll"
   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, thereby generating an error message.

   This line loads the `iplanet_web32.dll` module and the functions used by the plugin. Replace *`<install_path>`* with the path to `iplanet_web32.dll`.

   The default path for Windows is *`<install_path>`*`\bin\`.

4. Add this new line to `obj.conf`.

   ```
   Init fn=enforcer_init
   ```

   This line tells the server to execute the `enforcer_init` function at server startup.

5. Add these same lines to the `<Object name=default>` section.

   ```
   PathCheck fn="enforcer_check"
   Error fn=enforcer_content reason="ENFORCER_CONTENT"
   ```

   where:

   — The first line must be listed as the first `PathCheck` directive.

   — The last line is used to display dynamic content to users.

6. Open `magnus.conf`.

7. Change the value of the `StackSize` parameter to `393216`. This prevents fatal errors from occurring in the Sun ONE (iPlanet) Enforcer plugin. If the default value is used, the Sun ONE

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

(iPlanet) Enforcer plugin runs out of stack—especially when logging to a Secure Audit server.

8. Restart your Web server.

To configure your server on Unix

1. Open `obj.conf`.

   By default, this file is located in `/usr/netscape/server4/https-<server name>/config/` where `<server name>` is the name of your server.

2. Locate the following line in `obj.conf`:

   ```
   Init fn=load-types mime-types=mime.types
   ```

3. Add the following line after it:

   ```
   Init fn="load-modules" shlib=<PATH><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.

   This line loads the Sun ONE (iPlanet) Enforcer plugin module and the functions used by the plugin. Replace `<PATH>` with the path to the Sun ONE (iPlanet) Enforcer plugin module and `<module>` with the filename (either `iplanet_web.so` on Linux and Solaris or `iplanet_web.sl` on HP-UX).

   The default path is `<install_path>/bin`.

4. Add this new line to `obj.conf`.

   ```
   Init fn=enforcer_init
   ```

   This line tells the server to execute the `enforcer_init` function at server startup.

5. Add these new lines to the `<Object name=default>` section.

   ```
   PathCheck fn="enforcer_check"
   Error fn=enforcer_content reason="ENFORCER_CONTENT"
   ```

   where:

   – The first line must be listed as the first `PathCheck` directive.

   – The last line is used to display dynamic content to users.

6. Open `magnus.conf`.

7. Change the value of the `StackSize` parameter to `393216`. This prevents fatal errors from occurring in the Sun ONE (iPlanet) Enforcer plugin. If the default value is used, the Sun ONE (iPlanet) Enforcer plugin runs out of stack—especially when logging to a Secure Audit server.

8. Restart your Web server.

Manually configuring the Sun ONE v6.0 Web server

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 Sun ONE (iPlanet) Enforcer plugin, edit your `magnus.conf` and `obj.conf` files 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.

Due to the dissolution of the relationship between Netscape and Sun, iPlanet Web servers have been renamed to Sun ONE. Note that this name change affects the name of the Enforcer plugin for this server. The Sun ONE (iPlanet) Enforcer plugin still supports existing iPlanet Web servers as well as the newly named Sun ONE v6.0 Web servers.

To configure your server on Windows

1. Open `obj.conf`.

   This file is located, by default, in:

   ```
   <installation root>\iPlanet\servers
   https-<server name>\config\
   ```

   where:

   — *`<installation root>`* is the installation location of your server

   — *`<server name>`* is the name of your server.

2. Locate the following line in `obj.conf`:

   ```
   NameTrans fn=document-root root="$docroot"
   ```

3. Add this line to `obj.conf` after the line shown in step 2:

   ```
   PathCheck fn="enforcer_check"
   ```

   This line is a `PathCheck` directive.

4. Locate the following line in `obj.conf`:

   ```
   AddLog fn=flex-log name="access"
   ```

5. Add this line to `obj.conf` after the line shown in step 4:

   ```
   Error fn="enforcer_content" reason="ENFORCER_CONTENT"
   ```

   This line is used to display dynamic content to users.

6. Save changes and close `obj.conf`.
7. Open `magnus.conf`.
8. Locate the following line:

   ```
   Init fn=load-types mime-types=mime.types
   ```

9. Add the following line after it:

   ```
   Init fn="load-modules" shlib=<PATH>\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.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

This line loads the Sun ONE (iPlanet) Enforcer plugin module and the functions used by the plugin. Replace *`<PATH>`* with the path to the `iplanet_web32.dll`.

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.

The default path is `C:\Program Files\HP OpenView\Select Access\bin`

10. Add this new line:

    ```
    Init fn=enforcer_init
    ```

    This line tells the server to execute the `enforcer_init` function at server startup.
11. Change the value of the `StackSize` parameter to `393216`. This prevents fatal errors from occurring in the Sun ONE (iPlanet) Enforcer plugin. If the default value is used, the Sun ONE (iPlanet) Enforcer plugin runs out of stack—especially when logging to a Secure Audit server.
12. Restart your Web server.

To configure your server on Unix

1. Open `obj.conf`.

   On Unix, this file is located, by default, in:

   ```
   /usr/iplanet/servers/https-<server name>.com/
   config/
   ```

   where *`<server name>`* is the name of your server.
2. Locate the following line in `obj.conf`:

   ```
   NameTrans fn=document-root root="$docroot"
   ```
3. Add this line to `obj.conf` after the line shown in step 2:

   ```
   PathCheck fn="enforcer_check"
   ```

   This line is a `PathCheck` directive.
4. Locate the following line in `obj.conf`:

   ```
   AddLog fn=flex-log name="access"
   ```
5. Add this line to `obj.conf` after the line shown in step 4:

   ```
   Error fn="enforcer_content" reason="ENFORCER_CONTENT"
   ```

   This line is used to display dynamic content to users.
6. Save changes and close `obj.conf`.
7. Open `magnus.conf`.
8. Locate the following line:

   ```
   Init fn=load-types mime-types=mime.types
   ```

9. Add the following line after it:

```
Init fn="load-modules" shlib=<PATH><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, thereby generating an error message.

This line loads the Sun ONE (iPlanet) Enforcer plugin module and the functions used by the plugin. Replace *<PATH>* with the path to Sun ONE (iPlanet) Enforcer plugin module and *<module>* with the filename (either `iplanet_web.so` on Linux and Solaris or `iplanet_web.sl` on HP-UX).

The default path is: `/opt/OV/SelectAccess/bin`

ℹ 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.

---

10. Add this new line:

    ```
    Init fn=enforcer_init
    ```

    The second line tells the server to execute the `enforcer_init` function at server startup.
11. Change the value of the `StackSize` parameter to `393216`. This prevents fatal errors from occurring in the Sun ONE (iPlanet) Enforcer plugin. If the default value is used, the Sun ONE (iPlanet) Enforcer plugin runs out of stack—especially when logging to a Secure Audit server.
12. Restart your Web server.

Configuring the Apache Web server

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 Enforcer plugin, edit your `httpd.conf` file to load the Enforcer module and several functions.

To configure the server 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:

⚠ You must add `mod_enforcer` lines after `mod_ssl`. This ensures that the Apache Enforcer plugin starts after the SSL module has already loaded.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

- **For Linux and Solaris:**

```
AddModule enforcer_module /opt/OV/SelectAccess/bin/mod
_enforcer.cpp
LoadModule enforcer_module /opt/OV/SelectAccess/bin/mod_
enforcer.so
```

- **For HP-UX:**

```
AddModule enforcer_module /opt/OV/SelectAccess/bin/mod
_enforcer.cpp
LoadModule enforcer_module /opt/OV/SelectAccess/bin/mod_
enforcer.sl
```

If your site is using its own access control handler, place it above the `enforcer_module` line in the `LoadModule` section.

If you intend to use SSL with your Apache Enforcer plugin, you need to also add the following line to your `httpd.conf` configuration file:

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

This line allows the plugin to export SSL data the Policy Validator requires. Without adding this line, any SSL encryption decision points in your existing rules fail.

4. Comment out the `ClearModuleList` line.
5. 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, see either *To start the Apache Web server on HP-UX manually* on page 172 or *To allow the Setup Tool to start the Apache Web server* on page 172 in the *HP OpenView Select Access 6.0 Installation Guide*.

To configure the server for the IBM HTTP server distributed with WebSphere

1. Make sure the Policy Validator is running.
2. Open your `httpd.conf` configuration file:

   ```
   C:\WebSphere\conf\httpd.conf
   ```

3. Add the following line to the beginning of the `LoadModule` section:

   ```
   LoadModule enforcer_module "C:\Program Files\HP
   OpenView\Select Access\bin\apache_web32.dll"
   AddModule enforcer_module "C:\Program Files\HP
   OpenView\Select Access\bin\mod_enforcer.cpp"
   ```

If your site is using its own access control handler, place it above the `enforcer_module` line in the `LoadModule` section.

4. Save `httpd.conf` and restart the IBM HTTP server distributed with WebSphere.

Other steps to take to secure your resources

Setting up an Enforcer plugin is not just limited to installing the plugin, configuring it, and then modifying the Web server's configuration file. It also requires that additional steps be taken following the configuration.

To Enforcer-protect the content of your Web server:

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 *Building the Resources Tree* on page 26 in the *HP OpenView Select Access 6.0 Policy Builder Guide*.
2. Set the access policy for this Web server.

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

3. 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 used.

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.

Configuring your Enforcer plugin for virtual hosting

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. 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.

Depending on the specific hosting environment, host organizations may not want to incur the cost on performance that occurs when you

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

Select Access-protect a Web server and consequently all virtual domains.

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 C, *Troubleshooting*, in the *HP OpenView Select Access 6.0 Policy Builder Guide*.

In addition to using IP addresses or host domain names, you can also use host header-based virtual host names on Apache, IIS, and Sun ONE servers.

Creating a pass-through domains list

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 *To set up a list of pass-through domains* on page 159 of the *HP OpenView Select Access 6.0 Installation Guide*.

---

iPlanet 4.0 Web server cannot handle virtual domains. For details on configuring this Web server, see *Manually configuring the iPlanet 4.0 Web server* on page 83.

Scenario: Virtually Yours hosting service

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

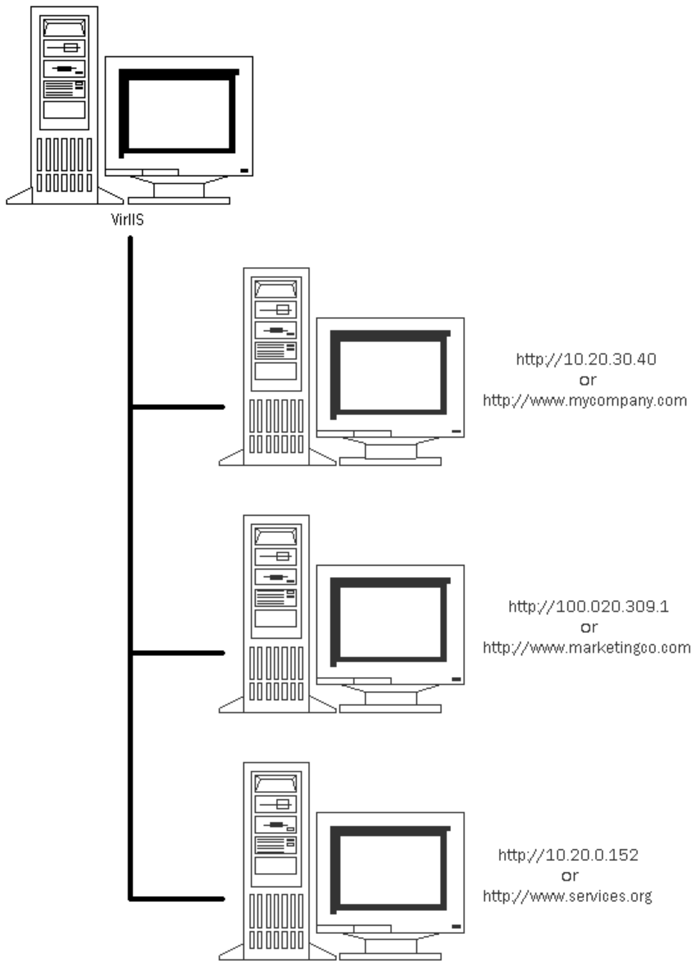


Figure 20: Example Web server with virtual domains

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 VirIIS' IIS Enforcer plugin with a **Pass-through Domain** list that might only include the following names:

```
marketingco
services
```

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.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

Using custom Web server plugins to include site-specific data

You can create your own custom plugins to add site-specific data to an XML query. Locate example `site_data` plugins for all three Web servers in `<install_path>\source`. To create a Web server-specific plugin, you must include the following calls:

- For iPlanet and Sun ONE Web servers:
  ```
  pblock_nvinsert("site_data", "value", rq->vars)
  ```
- For Apache Web servers:
  ```
  ap_table_add(r->notes, "site_data", "value")
  ```
- For IIS Web servers:
  ```
  pHeaders->AddHeader(pFC, "site_data", "value")
  ```
  where `pFC` is a pointer to `HTTP_FILTER_CONTEXT`

This custom plugin must be configured to execute before Select Access's Enforcer plugin, so that details that it does not normally support can be extracted and forwarded to the Policy Validator for further processing. Examples of these plugins are included on the Select Access CD in the `source` folder. The examples show how to use a custom plugin to extract time data. You can modify these example plugins accordingly to include whatever site-specific data you require.

Select Access forms used by Enforcer-protected Web servers

Select Access includes several forms that are used to support Select Access's key features. By default, these forms are installed in the `<install_path>/content` folder. Before using these forms, you may want to understand the following topics:

- *Customizing forms:* If your user are performing any kind of form-based login, you can customize the form templates used by the Enforcer plugin. For details, see *Customizing Select Access support forms* on page 94.
- *Avoiding the loss of data over a POST*: The Enforcer plugin can preserve POSTed data when the destination of that POST is to a protected resource where access is restricted. For details, see *Preserving POST data* on page 95.
- *Form behavior*: When updating from a previous version of Select Access to Select Access 6.0, you must ensure that you update Select Access's forms. Forms from previous versions are missing a hidden refresh parameter that is critical to the forms' behaving correctly with this new version. Otherwise, when a form is filled in and then submitted, the user cannot be redirected to the default page. Instead an error message is displayed. For details on how to update your forms correctly, see *Getting the correct form behavior* on page 96.

Customizing Select Access support forms

Customizing these form templates requires that you:

1. Modify the template to apply enterprise-specific styles and standards by:
   - Modifying the text
   - Changing the layout
   - Adding graphics

You cannot modify certain form elements. For details on where and how you can customize these forms, see the corresponding form-specific section that follows.

2. Save the form. You can save it to the same default filename, or you can give the form a new name. However, if you change the file's name, remember to update the corresponding Policy Builder plugin's configuration.

   For example, if you save `Radius_form.html` as `Radius_authentication.html`, you must change the path to the form in the **RADIUS Server Properties** dialog.
3. Ensure that all customized forms have been saved to the corresponding `<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.

Table 27 summarizes the categories of template forms included with Select Access.

Table 27: Types of template forms available

Support forms used by...	For details, see...
Logging into a Select Access-protected system.	*Customizing standard login form templates* on page 97
User-driven password management.	*Customizing password management form templates* on page 99
Authentication servers and their specific authentication method.	*Customizing the form-based authentication templates* on page 101
User-driven profile management.	*Customizing the Profile Management form templates* on page 107

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

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, information is not lost as the user is prompted for authentication.

If you have custom files, the installer backed up these forms to a subfolder of the `<install_path>/content/` folder. If you want to duplicate the functionality added in these new forms to your existing one, you need to manually update them accordingly. For details or to see a complete list of files that require this change, see *Getting the correct form behavior* on page 96.

The Enforcer plugin evaluates the initial request to see:

- *If the request was a GET*: If so, the Enforcer plugin behaves as it did in the unpatched version of Select Access 6.0.
- *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 user's 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 user.
  - If JavaScript is disabled in the user's browser, the Enforcer plugin displays the form with a **Submit** button, which allows the user to manually trigger it.

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

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.

Testing the plugins new 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.

Getting the correct form behavior

When updating from a previous version to Select Access 6.0, 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 patch 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 user cannot be redirected to the intended page. Instead, an error message is displayed.

To update your custom forms

1. Open a customized form.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

2. Add the hidden refresh parameter anywhere between the `<FORM>` and `</FORM>` tags:

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

3. 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%%">
   ```

4. 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 user.

5. Repeat steps 1-4 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.

6. When you have overwritten all the files in the `<install_path>\content` folder, delete the `content_prepatch3` subfolder.

Customizing standard login form templates

There are certain forms that tell users whether or not they have been authenticated based on the authentication information they have provided. Table 28 lists these forms.

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 288 in the *HP OpenView Select Access 6.0 Policy Builder Guide*.

Table 28: Authentication form templates

This form...	Is used by this Enforcer plugin...
`accepted.html`	*All* Enforcer plugins. Form-based authentication of any kind needs it (that is, registration and RADIUS authentication). The Enforcer plugin uses this form to notify an authenticated user that access to the resource they have requested is pending.
`deny.html`	*IIS Enforcer plugin* only, because the IIS Web server does not have its own default deny page.
`login_form.html`	*All* Enforcer plugins. This login form replaces native browser authentication (also known as basic authentication). The login form collects user credentials (username and password) on a separate form, in order to prevent the information from being cached in the Web browser. Using this form: • Allows the Policy Validator to terminate a user session • Prevents unauthenticated users from gaining access to information and resources via browsers that have been left open

Modifying the accepted and deny pages

You can customize these forms by adding or modifying text and graphics. You do not need to worry about accidentally deleting important template components as these pages are strictly text-based.

You cannot change the names of these files. The Enforcer plugins are programmed to look for these pages by name. If you alter the name in any way, Enforcer-protected Web servers cannot display these pages to the user.

Modifying the login form

You can customize this form by adding or modifying text and graphics. However, when modifying `login_form.html`, do not change the names or values for the form elements:

- The first field in the form must be:
  `<INPUT TYPE="text" NAME="user" VALUE="">`

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

- The second field must be `<INPUT TYPE="password" NAME="password" VALUE="">`
- When a user logs in, the information must be submitted using: `<INPUT TYPE="submit" NAME=".submit" value="Login now">`
- You also need to 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.

Optionally, you can delete the following line, if you do not want your users to know what password server or realm they are authenticating against: `Password Server: %%form_realm%%`

Customizing password management form templates

There are two distinct categories to the form templates that support password management functionality:

- The *required management forms* category allows the user to supply existing credentials (username and password) as well as submit a new password. Copy these forms to your Web server:
  `password_change_form.html password_change_unable_form.html password_confirm_mismatch_form.html password_missing_field_form.html password_userid_form.html`
- The *optional error message forms* category that support the accurate input of the password management fields. Depending on the password policy that exists, any number of these error messages are required.

Forms that support password management are listed in Table 29. For details on setting up password management, see Chapter 10, *Managing user accounts*, in the *HP OpenView Select Access 6.0 Policy Builder Guide*.

Table 29: Password management form templates

If your password policy includes...	You need this form...
Matching words from the dictionary file you configured during the Policy Validator's setup. For details, see Chapter 7, *Configuring the Policy Validator*, in the *HP OpenView Select Access 6.0 Installation Guide*.	`password_dictionary_match_form.html`
Maximum password age and a warning that the password expires.	`password_expired_form.html` and `password_expiry_form.html`

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

Table 29: Password management form templates (Continued)

If your password policy includes...	You need this form...
Match against a specified number of a user's previous passwords.	`password_history_match_form.html`
Minimum and/or maximum password lengths.	`password_invalid_length_form.html`
Password uniqueness that forces the user to include alphanumeric character types within the user's password.	`password_missing_chars_form.html`
Matching against a user's given name, family name, user ID, or CN.	`password_username_match_form.html`
Allows the user to update passwords.	`password_change_form.html`

Modifying the password management and error forms

Most of the password management form templates are the same except for the error message that precedes the form fields in those templates. As a result, the rules governing the customization of these forms are the same: you can customize these forms by adding or modifying text and graphics. However, when modifying either of these forms, do not change the names or values for the form elements:

- The first field in the form 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="">`
- The fifth field must be:`<INPUT TYPE="submit" NAME="password_mgmt" value="Change Password">`
- The last lines contain the hidden attributes used by the form elements and are required by Select Access: `<INPUT TYPE="HIDDEN" NAME="form_realm" VALUE="%%form_realm%%"/>`

Modifying the password expiry warning form

The Enforcer plugin uses the `password_expiry_form.html` form to warn 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 user chooses to change their password now, the Enforcer plugin displays `password_change_form.html`.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

You can customize this form by adding or modifying text and graphics. However, when modifying either of these forms, do not change the names or values for the form elements:

- You must not modify the `%%days_remaining%%` text substitution placeholder. It is used to dynamically insert the correct number of days remaining before the user's password expires.
- The first field in the form must be: `<INPUT TYPE="submit" NAME="password_change" value="Change Password Now">`.
- The second field in the form must be: `<INPUT TYPE="submit" NAME="password_defer" value="Change Password Later">`.

Customizing the form-based authentication templates

There are two kinds of form-based authentication templates. These forms are described in Table 30. For details on setting up the authentication servers that require the Enforcer plugin to display these forms, see Chapter 6, *Setting up authentication servers*, in the *HP OpenView Select Access 6.0 Policy Builder Guide*.

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 288 in the *HP OpenView Select Access 6.0 Policy Builder Guide*.

Table 30: Authentication form templates

This authentication method...	**Requires this form...**
Registration	`registration_form.html`: Enforcer plugins use this form to collect information on users. The process of registration can change the user from an unknown user to a known user. This allows administrators to set explicit policies for them, rather than just inheriting the unknown user policy. For details on how to modify this form, see *Modifying the registration form* on page 103. **Note:** For eTrust directory server, the fax and telephone number fields located on the registration server form are mandatory. HP recommends you modify the `registration_form.html` form by placing an asterisk beside the fields to indicate that they are mandatory.
Multiple server authentication	`multiauth_form.html`: When more than one authentication server is configured for a single resource, Enforcer plugins uses the `multiauth_form.html` template in SelectID as in a rule. This template contains four elements: • Password • SecurID • RADIUS • Registration • NTLM/Kerberos For details on using SelectID, see Chapter 6, *Setting up authentication servers*, in the *HP OpenView Select Access 6.0 Policy Builder Guide*. For details on how to modify this form, see *Modifying the ordering of authentication servers template* on page 104.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

Table 30: Authentication form templates (Continued)

This authentication method...	Requires this form...
RADIUS authentication server	`radius_form.html`: Enforcer plugins use this form to present challenges to and capture their responses from users. For example: 1. The user requests a network resource. 2. The request is transferred as a query to the Enforcer plugin. 3. The Enforcer plugin forwards the query to the Policy Validator. 4. The Policy Validator then passes the query to the RADIUS server. 5. The RADIUS server issues a challenge back to the Policy Validator. 6. Policy Validator forwards the challenge to the Enforcer plugin. 7. The Enforcer plugin uses this form to present the challenge, capture the response from the user, and forward the information to the RADIUS server via the Policy Validator. 8. At this point the RADIUS server either issues an ALLOW or a DENY, or even another challenge (at which point the process repeats itself). For details on how to modify this form, see *Modifying the registration form* on page 103.
Integrated Windows, NTLM, and Kerberos authentication servers	`winauth_form.html`: Only IIS Enforcer plugins use this form to replace native browser authentication (also known as basic authentication). The login form collects user credentials (username, password, and domain name) on a separate form, in order to prevent the information from being cached in the Web browser. Using this form: • Allows the Policy Validator to terminate a user session • Prevents unauthenticated users from gaining access to information and resources via browsers that have been left open For details on how to modify this form, see *Modifying the Integrated Windows, NTLM, or Kerberos form* on page 106.

Modifying the registration form

You can customize this form by adding or modifying text and graphics. However, when modifying `registration_form.html`, do *not* change the names or values for the form elements in the following sections of the registration form.

First time registration section:

- The first field in the form must be:
  `<INPUT TYPE="text" NAME="givenName" VALUE="">`
- The second field must be:
  `<INPUT TYPE="text" NAME="sn" VALUE="">`
- When the registration area of the form is filled in, it must be submitted using:`<INPUT TYPE="submit" NAME=".submit" value="Register now">`
- You also need to include the following field somewhere on this page:
  ```
  <INPUT TYPE=”HIDDEN” NAME=”selectaccess_send_refresh”
  VALUE=”enable”/>
  ```

Already registered section:

- The login information must be submitted using:
  `<INPUT TYPE="submit" NAME=".login" value="Login now">`
- You also need to include the following field somewhere on this page:
  ```
  <INPUT TYPE-”HIDDEN” NAME=”selectaccess_send_refresh”
  VALUE=”enable”/>
  ```

The `selectaccess_send_refresh` 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.

Modifying the ordering of authentication servers template

You can customize the `multiauth_form.html` form in the following ways:

- Add text and graphics that maintain your company’s corporate look and feel.
- Add or delete sections of the form based on the authentication methods your company supports. Some companies only support certain authentication methods, while others write their own authentication plugins.
- Comment out or add fields.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

- Reorganize the sections of the form.

⚠ If a user logs in with credentials for which a deny policy is set, no message notifying the user of this policy is displayed. This can cause some confusion if your users think they had incorrectly entered their user credentials. It is a good idea to customize this form to notify users of this possibility.

By default, each section of the `multiauth_form.html` file contains the following two default fields:

- The first field in the password, SecurID, and RADIUS sections of the form must be: `<INPUT TYPE="text" NAME="user" VALUE="%%user%%">`.
- The second field in the password, SecurID, and RADIUS sections of the form must be: `<INPUT TYPE="password" NAME="password" VALUE="">`.

The registration section of the form contains the following additional fields:

- 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 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=””>
```

where,

- `TYPE` is always `”text”` because users only input ASCII data.
- `NAME=”attribute_name”` is the name of the additional attribute you activated.

Modifying the RADIUS form

You can customize this form by adding or modifying text and graphics that make the form more user-friendly. However, when modifying this form, do *not* change the hidden attributes for the form elements. For example:

- The first field in the form must be:
  ```
  <INPUT TYPE-”HIDDEN” NAME=”form_radius_state”
  VALUE=”%%form_radius_state%%”/>
  ```
- You also need to include the following field somewhere on this page:
  ```
  <INPUT TYPE-”HIDDEN” NAME=”selectaccess_send_refresh”
  VALUE=”enable”/>
  ```

Enforcer-protected Web servers require the `selectaccess_send_refresh` field, which 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.

Optionally, you can delete the `%%form_realm%%` line, if you do not want your users to know what RADIUS server or realm they are authenticating against.

Modifying the Integrated Windows, NTLM, or Kerberos form

You can customize this form by adding or modifying text and graphics that make the form more user-friendly. However, when modifying this form, do *not* change the hidden attributes for the form elements. For example:

- 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="">.`
- The third field must be `<INPUT TYPE="text" NAME="domain" VALUE="">.`
- When a user logs in, the information must be submitted using:
  `<INPUT TYPE="submit" NAME=".login" value="Login now">.`
- You also need to 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.

Optionally, you can delete the following line, if you do not want your users to know what password server or realm they are authenticating against: `Password Server: %%form_realm%%`

Customizing the Profile Management form templates

There are two distinct categories to the form templates that support profile self-management:

- The *profile self-management form* that allows users to modify their profiles. The default template contains the following attributes: first name, last name, email address, phone number, fax number, and user password. These attributes mirror the recommended attributes activated in *The profile self-management terminal point* on page 141.
- The *message forms* communicate the status of a user's updated profile.

For details on how to enable profile self-management, see Chapter 10, *Managing user accounts*, in the *HP OpenView Select Access 6.0 Policy Builder Guide*.

These forms are listed in Table 31.

Table 31: Profile management form templates

This form...	Is used by the Enforcer plugin when...
`profile_mgmt_form.html`	The user should update her profile.
`password_change_form.html`	The user should update her password.
`profile_error_form.html`	An error occurs while the user is trying to update her profile. For example, this error form would appear if there was a failed connection between the Policy Validator and LDAP.
`profile_no_user_form.html`	The user does not have a profile on the system.
`profile_ok_form.html`	The user's profile has been successfully updated.

Modifying the profile self-management form

You can customize the profile self-management and message forms by adding text and graphics that maintain your corporate look and feel.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

You can further customize the `profile_mgmt_form.html` file by commenting out or adding fields.

- 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.

By default, the `profile_mgmt_form.html` file contains the following five attributes. 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%%">`

By default, the `profile_mgmt_form.html` file also contains the `userPassword` attribute, which appears as the **Change Password** button at the bottom of the form. If you activate this attribute, when a user clicks the **Change Password** button, the `password_change_form.html` form appears. If you do not want users to change their own password, you need to remove the `userPassword` attribute from the list of activated attributes. You also need to comment out this button so that it does not appear on the `profile_mgmt_form.html`.

Because the user's password attribute (`userPassword, password_id`, etc.) can vary depending on which directory server she is using, you may need to change the attribute defined for the **Change Password button** in the `profile_mgmt_form.html` form.

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 users only input ASCII data.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

- `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.

Modifying the message forms

You can customize these forms by adding or modifying text and graphics. You do not need to worry about accidentally deleting important template components as these pages are strictly text-based.

You cannot change the names of these files. The Enforcer plugins are programmed to look for these pages by name. If you alter the name in any way, Enforcer-protected Web servers cannot display these pages to the user.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

Chapter 7

Using Select Access personalization information

Personalization is the process of generating or modifying dynamically generated pages to customize Web sites (Internet, intranet, and extranet) or Web Services (J2EE or .NET) for users, groups of users, or users belonging to a pre-defined role.

The contents of this chapter apply to Sun ONE (iPlanet), Apache, and IIS Web servers. For information on how to implement personalization for Domino, see the *Domino Integration Paper* on the product CD.

The contents of this chapter only outlines information on how to leverage the user attributes encoded in HTTP header variables. For information on how to enable personalization in Select Access, see Chapter 5, *Authentication fundamentals: SelectID and personalization* in the *HP OpenView Select Access 6.0 Policy Builder Guide*.

Server-side specifics

Because the user information used to personalize content is centralized on a directory server, the process of setting up a personalized network is greatly simplified: developers do not need to develop or maintain a complex directory with corresponding access code from which entitlements are extracted. Instead, the Select Access system passes this information directly from its user data source.

Setting up personalization on the server-side requires that you:

- Configure your server-side content by organizing personalized Web content that requires authentication, so that resources requiring similar access–control requirements are loaded from a corresponding page that acts as a launching point.

You do not want a single page loading all your content. Otherwise, you cannot use the Policy Matrix to set different rules for that content.

- Create the server-side scripts that extract and evaluate the user information embedded in the HTTP headers forwarded by the Policy Validator to the Enforcer plugin. For details, see *Extracting required user information* on page 112.

Extracting required user information

Once users are defined with their corresponding user entry, and the attribute names and values contained therein, the following Web objects can extract the information accordingly:

- CGI scripts
- ASP pages
- JSP program
- COM objects

Therefore, integrating the two sides of personalization on a Select Access-protected network in large part deals with how to extract this user information before making decisions on what content the user ultimately sees.

If you are exchanging user attributes between two Select Access SAML servers, you are limited in the types of attributes you can activate. For a list of supported attributes for SAML, see the *HP OpenView Select Access 6.0 SAML Solution Guide*.

What happens on Select Access

The sequence of Select Access behavior is summarized by the following steps:

1. Policy Builder records the attributes you activate to the Policy Store. Users only get access to personalized content when attribute values meet those defined by the administrator. For details on how to enable these personalization attributes, see *To enable personalization* on page 73.
2. The Policy Validator receives an authentication query, and downloads the personalization attributes from the Policy Store. These attributes are only included if the user has an allow policy set for the resource in question.
3. The Policy Validator then builds a reply to the Enforcer plugin with personalization data in XML. When an Enforcer plugin gets the reply, it exports these attributes as environment variables through HTTP headers that contain the XML data. For details, see

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

The structure of a Policy Validator's reply on page 113 of the *HP OpenView Select Access 6.0 Network Integration Guide*.

The Enforcer plugin is limited in the way it exports data. You can enhance the way you handle personalization if you write your own plugin.

4. The Web server then takes variables, decodes them, and displays the requisite content.

The structure of a Policy Validator's reply

The Policy Validator's reply contains XML attribute definitions that are structured as follows:

- *Multiple attribute name separation values*: An attribute of a user entry might have multiple values. All values of the attribute are exported to a single variable and are separated by a comma.
- *Multiple group and role name separation attribute values*: A user might be a member of one or more groups and/or roles. The names of all groups and roles are exported to a single variable and are separated by a comma.
- *Multiple groups and roles with multiple attribute values*: A user can be a member of one or more groups and/or roles. An attribute of a group or role can have multiple values. All values of the attribute for all the groups or roles are exported to a single variable. As a result, names and values are nested using the following format:
  - Attribute values are separated by a comma. For example, `xxx,y%2cyy`
  - Multiple groups and roles are separated by a semicolon. For example, `group1=xxx,y%2cyy;group2=group2`
  - Group and role names are separated from attribute values by the equal sign. For example, `group1=xxx`
- *Attribute values*: Because the comma, semicolon, equal sign, and percentage symbol are used as separators, they are URL-encoded if they are part of an attribute value, as illustrated by some examples in Table 32.

Attribute names are case insensitive. Binary attribute values are not supported.

Table 32: Example list of URL encoded characters

This character...	Is encoded to this...
`,`	`%2c`
`;`	`%3b`

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

Table 32: Example list of URL encoded characters (Continued)

This character...	Is encoded to this...
=	%3d
%	%25

Other characters are also encoded in order to protect the interaction between directory server and other Select Access components. If you are using the attributes for personalization, these characters must be first decoded. For details on how to URL decode these characters, see *Decoding encoded characters* on page 114.

An example of a typical Policy Validator response is provided in Code example 6.

Code example 6: Policy Validator response example

```
<PolicyValidatorReply>
<PROPERTY NAME="queryID">2</PROPERTY>
  <PROPERTY NAME="authenticated_dn">cn=Lance
Mountain,ou=customer,dc=com,dc=user_isp</PROPERTY>
  <PROPERTYLIST NAME="personalization">
    <PROPERTY NAME="User">lmountain</PROPERTY>
    <PROPERTY NAME="Phone">504%2D2325</PROPERTY>
    <PROPERTY NAME="Fax">504%2D2399</PROPERTY>
    <PROPERTY NAME="DName">
cn%3Dlance%20mountain%2Cou%3Dlance%2Cdc%3Dcom%2Cdc%3Duser_isp%2Cdc</PROPERTY>
    <PROPERTY NAME="Groups">customer%20people</PROPERTY>
  </PROPERTYLIST>
  <PROPERTY NAME="login_time">Thu Feb 27 12:59:45 2003
</PROPERTY>
  <PROPERTYLIST NAME="authentication_server_types">
    <PROPERTY NAME="authentication_method">password</PROPERTY>
  </PROPERTYLIST>
  <PROPERTY NAME="action">ALLOW</PROPERTY>
</PolicyValidatorReply>
```

Decoding encoded characters

When using data returned by the Policy Validator's reply, you need to unescape encoded characters that have been escaped prior to being displayed to the user. Decoding URL-encoded characters translates characters that have been escaped with the % character back into its recognizable state.

To decode URL-encoded strings

1. Split the variable into separate attribute values. All values of the attribute are separated by a comma, semicolon, equal sign, and percentage symbol.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

2. URL-decode the value. HTML-friendly languages like ASP have an `unescape()` function built into the language. Typically, the syntax of this function is:

   `unescape (<`*`string`*`>)`

   Use this function to access any personalization attributes you need.
3. Convert the decoded value from UTF-8 to the character set you want.

Enhancing personalization security

To enhance security for Select Access's implementation of personalization, Enforcer plugins can now export personalization attributes as variables by using the prefix `HTTP_SA_` instead of `HTTP_`. `HTTP_` remains the default prefix used.

Re-writing personalization prefixes

If you want to enable personalization in a more security-sensitive way, you must modify your Web server scripts to use this new prefix. With this prefix modification, the Apache and Sun ONE (iPlanet) Enforcer plugins can now check HTTP requests to see if they have been forged to inject personalization settings. If an Enforcer plugin determines that an HTTP header has been forged, it now automatically denies the request.

⚠ Because IIS cannot detect attempts to inject personalization settings, you should use the COM interface to access personalization attributes in a security-sensitive way. The COM interface is not susceptible to fraudulent header injections.

To secure personalization

To make Select Access's personalization implementation more secure, you must disable the backwards compatibility for personalization. This can be done using the Custom Settings wizard in the Select Access Setup Tool.

By disabling this backwards compatibility, you configure the Enforcer plugin to use the more security-conscious implementation of personalization.

To disable the backwards compatibility mode parameter, remove the **USE_OLD_P13N** flag from the **Custom Settings Flags** setup screen. For more information on using the Custom Settings setup wizard, see *Using the Setup Tool to configure the custom settings flags* on page 198 of the *HP OpenView Select Access 6.0 Installation Guide*.

IIS Web server and differences in personalization

Because of the way in which IIS has implemented server-side personalization, there are two key differences in the way in which personalization information is extracted. The two main differences are:

- The use of the ALL_HTTP header.
- The use of COM objects.

These differences are described in greater detail below.

Using the ALL_HTTP header

Due to the way environment variables on IIS export HTTP headers, variable names that contain an underscore (_) cause the value for the variable not to get exported to the corresponding HTTP header. As a result, ASP pages, CGI scripts, or JSP programs do not display values that are needed to personalize the output.

For example, if you configure your **Personalization** tab in the **Authentication Properties** dialog to export the `surname` attribute in an environment variable called `lastname` to an IIS Web server, the `HTTP_SA_LASTNAME` header that contains the exported XML value is empty. This can cause your Web site's personalization to break.

To work around this IIS issue with the underscore character, you can try either of these workarounds:

- Avoid using the underscore character. IIS headers without an underscore contain all exported user data correctly.
- If the underscore is required, use the `ALL_HTTP` header. This header contains all header names and their corresponding value in a single header element. Code example 7 shows an example of the `ALL_HTTP` header. All personalization variables appear bolded in this example.

To extract the required personalization information from the ALL_HTTP header, use a parsing tool.

Code example 7: Example ALL_HTTP header

```
ALL_HTTP=HTTP_ACCEPT:image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
application/vnd.ms-powerpoint, application/vnd.ms-excel, application/msword, */*
HTTP_ACCEPT_LANGUAGE:fr-ca HTTP_CONNECTION:Keep-Alive HTTP_HOST:tiny
HTTP_REFERER:http://www.mycompany.com/auth/ HTTP_USER_AGENT:Mozilla/4.0 (compatible;
MSIE 5.5; Windows NT 5.0)
HTTP_COOKIE:SelectAccess.PolicyBuilder=rO0XXXNyABNqYXZhLnV0aWwuSGFzaHRhYmxlE7sPJSFK5L
gDAAJGA***psb2FkRmFjdG9ySQAJdGhy%0D%0AZXNob2xkeHA%2FQAAAAAAACHcIAAAAwAAAcv0ABdXYXJuQ
2xlYXJWYWxpZGF0b3JDYWNoZXNyABFq%0D%0AYXZhLmxhbmcuQm9vbGVhbs0gcoDVnPruAgABWgAFdmFsdWV4
cAF4%0D%0A;
PolicyUser=AQAAAAAAPRHZhAAAAAA9EdvcdGlueS5jYS5iYWx0aW1vcmUuY29tOjk5OTkAcGFzczEAY249c3
RldmUgayxvdT1zdGV2ZSxkYz1jYSxkYz1iYWx0aW1vcmUsZGM9Y29tAGdhucvRWMv/rgd7M8YaKV529wKZop4
r25S88x5S50eUXv5LJbZ4lrOcHCEC/mJTjxQdm4FMNfHpXBaM1mqiUWljcjxfkwb6VUNUfSDQaR+VTcYQpSft
```

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

```
SISbigkf8T9yIV6oPUo7L8WLQqRKyTWdfXaW0WVABz1TctApuXOqLvpQ HTTP_ACCEPT_ENCODING:gzip,
deflate HTTP_SA_MMPF:19439 HTTP_SA_LASTNAME:Smith
HTTP_SA_DN:cn%3Djane%20s%2Cou%3Djane%2Cdc%3Dca%2Cdc%3Dmycompany%2Cdc%3Dcom
HTTP_SA_GROUPS:rd
```

To parse the ALL_HTTP header

1. Extract the `ALL_HTTP` header.
2. Use a parsing tool and split the `ALL_HTTP` header at each instance of "HTTP_SA_". This creates a list of headers and corresponding value pairs.

Using COM objects

On Windows, Web servers can access personalization attributes with COM (Component Object Model) objects because information is returned in standard XML. COM is particularly important to IIS Web servers, because they cannot detect attempts to inject personalization data into HTTP requests.

Use VBScripts to define personalization server-side details on ASP or CGI pages. For example, use the code shown in Code example 8 to access the personalization attributes the administrator configured in step 1. When you parse the COM objects, the results appear similar to that of Code example 9.

Code example 8: Personalization code for COM objects

```
<%
Set personalizer = CreateObject("SAIISPlugin.SAPersonalizer")
Set g = Request.ServerVariables("HTTP_SAMMPF")
personalizer.init(g)
x = personalizer.getXML
Response.Write x
%>
```

Code example 9: Parsed COM object

```
PROPERTY: cName -> Jane%20Smith
PROPERTY: phone -> 554%2D5555
PROPERTY: mail -> Jane%40mycompany%2com
PROPERTY: user_id -> JaneSmith
PROPERTY: dname -> cn%3DJane%20Smith%2Cou%3Djane%2Cdc%3Dca%2Cdc%3Dmycompany%2Cdc%3Dcom
PROPERTY: groups -> jane,mycompany%20people,management
```

Personalizing Web Services

You can use Select Access personalization data to personalize both J2EE and .NET Web Services. Both the Axis and the WSE Enforcer plugin sit in the path of Web Service SOAP requests sent by clients. Personalization data from the Policy Validator response can be made available to other components involved in processing a request.

Extracting personalization information from the Axis Enforcer plugin

If the reply from the Policy Validator contains personalization information, the Axis Enforcer plugin processes the data into a property/value map, and saves this map as a property named `SA_PERSONALIZATION` in the Axis MessageContext. The Axis engine makes this MessageContext available to the other components called while processing the request, including other Axis handlers and the Web Service itself.

Any component invoked after the Axis Enforcer plugin can be modified to extract personalization information. The following sample method shows how to extract the `SA_PERSONALIZATION` property from the MessageContext and print each name/value pair.

Code example 10: Extracting personalization information from the Axis Enforcer plugin

```
private void printSAPersonalizationData()
{
   MessageContext mc = MessageContext.getCurrentContext();
   Map map = (Map) mc.getProperty("SA_PERSONALIZATION");
   if ( map != null ) {
      System.out.println("SELECT ACCESS PERSONALIZATION DATA");
      Set keys = map.keySet();
      Iterator i = keys.iterator();
      while ( i.hasNext() ) {
         Object key = i.next();
         String str = (String) key + "=" + (String) map.get(key);
         System.out.println(str);
      }
      System.out.println("END OF SELECT ACCESS PERSONALIZATION DATA");
   }
}
```

Extracting personalization information from the WSE Enforcer plugin

If the WSE Enforcer plugin receives personalization data from the Policy Validator, it processes that data into a hashtable of name/value pairs and adds it to the HttpContext object that was created for the given request. The HttpContext object is made available to other components called while processing the request, including other WSE Input filters and the Web Service itself.

Any component invoked after the WSE Enforcer plugin can extract the Select Access personalization data using the `SA_personalization_attributes` key. The following sample method extracts the personalization data from the HttpContext object and prints a string containing the Select Access personalization name/value pairs.

Code example 11: Extracting personalization information from the WSE Enforcer plugin

```
public string PrintPersonalization()
{
   string personalizationData = "SA PERSONALIZATION DATA:\n";
   if ( HttpContext.Current.Items.Contains("SA_personalization_attributes") )
```

```
{
   IDictionary dict = HttpContext.Current.Items;
   Hashtable table = (Hashtable) dict["SA_personalization_attributes"];
   foreach (DictionaryEntry entry in table)
   //foreach (string name in table.Keys)
   {
      personalizationData += entry.Key;
      personalizationData += " = ";
      personalizationData += entry.Value;
      personalizationData += "\n";
   }
}
return personalizationData;
```

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

Chapter 8

Enabling single sign-on

Single sign on (SSO) is a key technology required for distributed computing. For example, a single username and password—once authenticated by the Policy Validator—acts like a passport, giving users access to distributed Web content, groupware, workflow or client and server applications. It allows the user to seamlessly move from one site to another without requiring reauthentication with each new site.

SSO requires that the Web server's authentication methods and server names match. This is necessary because when the Policy Validator generates a cookie for the authenticated user, it includes a delimited list of authentication server names. As the user moves from one server to another, the authentication server name and the authentication method for that server must be of the same level. If not, SSO fails and the user is required to reauthenticate with the new Web server. For details on authentication servers, see Chapter 6, *Setting up authentication servers*, in the *HP OpenView Select Access 6.0 Policy Builder Guide*.

Select Access supports three different levels of SSO, each of which requires a different configuration to enable it on your Select Access-protected network:

- Single domain SSO: A single-domain network environment is one where multiple Web servers with unique resources exist on a single cookie domain. For details, see *Configuring SSO on single Internet domains* on page 122.
- Multiple domain SSO: A multidomain network environment is one where multiple Web servers with unique resources exist on more than one cookie domain. For details, see *Configuring SSO on multiple Internet domains* on page 123.
- SSO across partnering organizations: A multidomain network environment that uses affiliations is one that requires the Security Assertions Markup Language (SAML) protocol. SAML provides an interoperable mechanism for passing credentials and other related information between Web sites that each have their own authentication and authorization system. For details, see the *HP OpenView Select Access 6.0 SAML Solution Guide*.

Configuring SSO on single Internet domains

You can set up multiple Enforcer-protected Web servers on a single cookie domain, so that users can access resources on them, without being required to reauthenticate each time or to provide additional user information.

For example, if a user is authenticated as JDoe at www.mycompany.com (Server1), and a hyperlink sends her to extranet.mycompany.com (Server2), she is never challenged by any of the Enforcer-protected Web servers because they share the same cookie domain. Figure 21 illustrates this sample scenario.

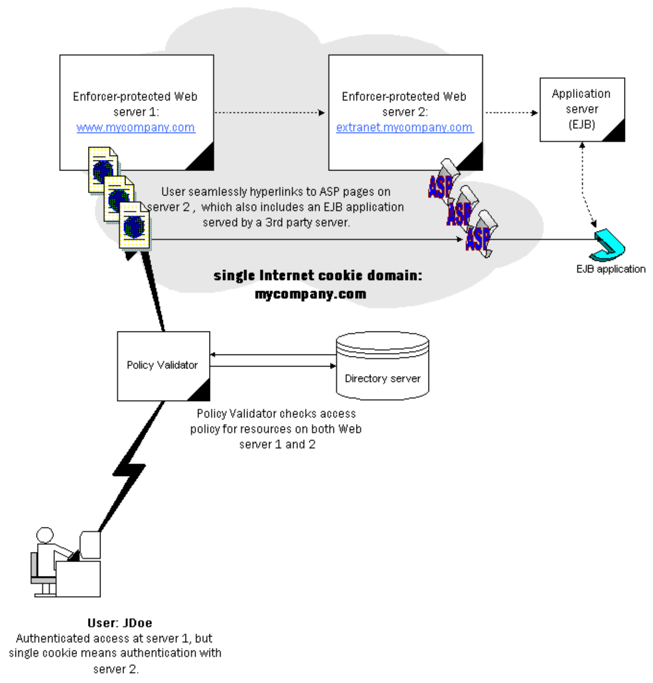

Enforcer-protected Web server 1: www.mycompany.com
Enforcer-protected Web server 2: extranet.mycompany.com
Application server (EJB)
User seamlessly hyperlinks to ASP pages on server 2 , which also includes an EJB application served by a 3rd party server.
ASP
EJB application
single Internet cookie domain: mycompany.com
Policy Validator
Directory server
Policy Validator checks access policy for resources on both Web server 1 and 2
User: JDoe
Authenticated access at server 1, but single cookie means authentication with server 2.

Figure 21: Single-domain SSO scenario

To configure SSO on a single domain only

1. When configuring your Enforcer plugin, perform a **Custom** configuration.
2. On the Enforcer plugin's **Single DNS domain SSO** setup screen, enter the cookie domain.
3. Repeat these steps on all Enforcer-protected Web servers.

For details, see Chapter 8, *Configuring the Enforcer plugins*, in the *HP OpenView Select Access 6.0 Installation Guide*.

Configuring SSO on multiple Internet domains

Using multidomain SSO is valuable for corporate networks with many Web servers with multiple domains because it makes authentication consistent and streamlined:

- User ID and password across all domains are synchronized through Select Access-protected networks.
- Fewer User ID and password pairs are needed.
- User ID and password pairs are centrally located.

A typical multidomain network environment can be an expensive proposition in terms of the administration effort that needs to be maintained as a user shifts across different domains. Credentials do not just need to be entered, they also need to be remembered. For each new set, there is an added security risk because users have the tendency to:

- Keep passwords the same so they are easy to remember.

  OR
- Make passwords unique but record them in an unsecured way.

To alleviate the symptoms of administration overhead and security risk, you can set up multiple Enforcer-protected Web servers on multiple cookie domains, so that users can access resources on them, without being required to reauthenticate each time or to provide additional user information.

For example, if a user is authenticated as JDoe at www.mycommercesite.com (Server1), and a hyperlink sends her to coolproduct.affiliateco.com (Server2) before stopping at canada.purchasemenow.com (Server3), she is never challenged by any of the Enforcer-protected Web servers on these three domains. Figure 22 illustrates this sample scenario.

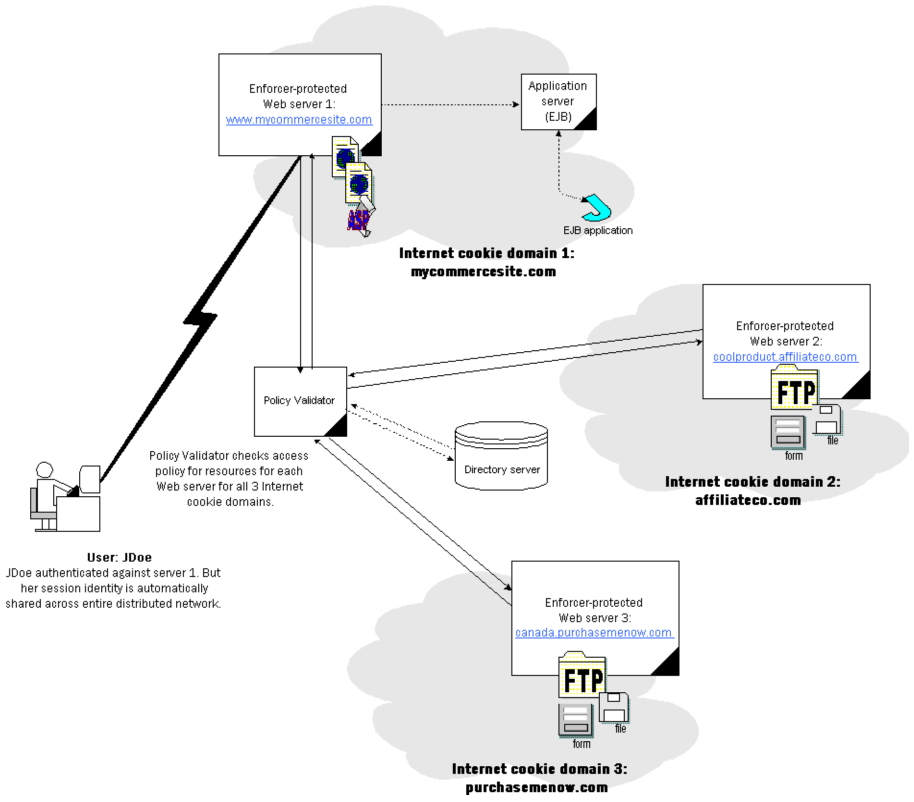

Enforcer-protected Web server 1: www.mycommercesite.com
Application server (EJB)
EJB application
Internet cookie domain 1: mycommercesite.com
Enforcer-protected Web server 2: coolproduct.affiliateco.com
FTP
file
form
Internet cookie domain 2: affiliateco.com
Policy Validator
Directory server
Policy Validator checks access policy for resources for each Web server for all 3 Internet cookie domains.
User: JDoe
JDoe authenticated against server 1. But her session identity is automatically shared across entire distributed network.
Enforcer-protected Web server 3: canada.purchasemenow.com
FTP
file
form
Internet cookie domain 3: purchasemenow.com

Figure 22: Multidomain SSO scenario

How the Enforcer plugin supports multidomain SSO redirects

With multidomain SSO, the user's browser is the central processing point for all communications: cookies are set and redirect requests are distributed to and from Web servers.

1. If a user is denied access on a Web server, and if the previous Web server is included in the protected domains list (for details, see *Setting up multidomain SSO with Select Access* on page 125), the user needs to be reauthenticated with a special URL that is redirected back to the originating server.
2. The previous Web server recognizes this URL, and redirects the browser to the new Web server with the cookie that authenticates the user on the second server.

Setting up multidomain SSO with Select Access

To ensure that a user accessing content on Enforcer-protected domains is not unnecessarily rechallenged for information, refer to the steps in Table 33:

Table 33: Multidomain SSO overview

This step...	For details, see...
Run the Setup Tool for each Enforcer plugin, and create a matching list of domains, so that they all share a mutually inclusive list of protected domains that allows analogous multidomain SSO: 1. When configuring your Enforcer plugin, perform a **Custom** configuration. 2. On the Enforcer plugin's **Multiple DNS domain SSO** setup screen, enter the cookie domain. 3. Repeat these steps on all Enforcer-protected Web servers.	*To set up multidomain single sign-on* on page 151 of the *HP OpenView Select Access 6.0 Installation Guide*
4. Register all Policy Validators with the same directory server. This is typically the same directory server that the Administration server uses as its Policy Store.	*How rogue Policy Validators are detected* on page 125
5. Ensure distributed Web servers use the same authentication server with the same authentication method.	*Configuring authentication methods* on page 126

Due to the way Internet Explorer uses the Referer headers when linking to a non-protected (HTTP) page from a protected one (HTTPS), multidomain SSO does not get triggered. For details, see Appendix C, *Troubleshooting*, of the *HP OpenView Select Access 6.0 Policy Builder Guide*.

How rogue Policy Validators are detected

All Web servers need to get their information from the same LDAP database, so that they can validate the nonces the Policy Validator generates. Because the nonce encodes user-specific information, it must be decoded by a Policy Validator that is able to look up data on the same database. There are two things the Policy Validator looks for:

- User-specific information to validate user data. The user data does not need to be on the same directory server as the Policy Store.
- Policy Validator-specific information to ensure that the Policy Validator that generated the nonce is not a rogue validator. Only Policy Validators registered in the Policy Store are considered valid directory servers. Policy Validators are automatically registered when you install and configure them with Setup Tool.

If Policy Validators cannot validate a cookie, the user is denied access and is forced to reauthenticate.

Configuring authentication methods

If you are using multidomain SSO, set up the same authentication method(s) *and* authentication server(s) for resources in the Policy Matrix. This is important because the cookie contains information on the authentication server that validated the user. If the server information in that cookie does not match the authentication server used for the new Web site the user is requesting, the user is forced to reauthenticate. For example, if a user is authenticated with password authentication on PassServ1, but moves to another part of your distributed network that uses password authentication on PassServ4, the user is required to reauthenticate, despite your efforts of setting up multidomain SSO.

However, there may be times when you want to tier the level of authentication used: the more confidential the distributed content is, the stronger the authentication you likely want to use. This method of setting up authentication across a distributed network is called "step-up authentication", and is something you may want to consider for your network.

For details on configuring your pool of authentication methods and servers, see Chapter 6, *Setting up authentication servers*, in the *HP OpenView Select Access 6.0 Policy Builder Guide*.

Understanding nonces and cookies

Since HTTP is a stateless protocol (that is, each transaction is completely separate from all others), HTTP cookies were devised as a method of preserving state between HTTP requests. This is done so

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

that you can keep track of shopping baskets, viewing preferences, and so on.

Do not try to decode the contents of Select Access's nonce to extract user information. This nonce is an internal data format that is subject to change without notice. If you want to extract user information for personalization or customization purposes, you can use personalization attributes instead. For details, see Chapter 7, *Using Select Access personalization information*.

Identifying users without reauthentication

Nonces and cookies are used by Select Access to help identify users as they navigate through one or more Web servers. While both elements contain user-specific information, only the cookie is set on the user's Web browser so the user can be identified without reauthenticating.

Nonces: issued by the Policy Validator and sent to the Enforcer plugin

A nonce is an opaque piece of data. Select Access's version of a nonce is typically translated into browser cookies. Select Access nonces can contain encrypted data such as:

- The user's DN
- The name of the Policy Validator that issued it
- The ID of the authentication servers used to authenticate the user's identity
- The time of user login
- The expiry date of the nonce
- A digital signature that ensures the nonce has not been tampered with

Cookies: issued by the Enforcer plugin and set in the browser

Cookies are the HTTP-specific way of sending the nonce that the Policy Validator issues. The Enforcer plugin takes the nonce and packages it with additional information before allowing the Web server to set it in the user's Web browser. The browser then sends this cookie in its HTTP headers (an example is shown in Code example 12) for each new content request it makes on behalf of the user.

Code example 12: HTTP reply header

```
Set-Cookie: PolicyUser=bWFub3dhci5jYS5iYWx0aW1vcmUuY29tOjk5ODh8dWlkPXN
rLG91PXVzZXJzLG91PXN0ZXZlLG89Y2EuYmFsdGltb3JlLmNvbXxMfHBhc3NfdmFsaWRhdG9yfDtM3z4FzLNl
wtEpKyGGtyl2k7OehNCdCQf2IcW+GV9gdn5n1jFYPsE21h/nCuHtt38n6sXk1lhBGv7t6qI18Rf13hnMg1Fq0
qCZjUMO6rXzfESetBO6NL1zF/wCIaM0r7oilummeK6gYVWukFwz3W+t2OYnNnJk5nPzPPozI49RRvaNDA/;
path=/; domain=.mycompany.com;
```

A user's browser returns a similar string to the Web server each time he tries to open a new session on that server, as illustrated in Code example 13.

Code example 13: Browser new session attempt

```
PolicyUser=Cookie:bWFub3dhci5jYS5iYWx0aW1vcmUuY29tOjk5ODh8dWlkPXNrLG91PXVzZXJzLG91PXN
0ZXZlLG89Y2EuYmFsdGltb3JlLmNvbXxMfHBhc3NfdmFsaWRhdG9yfDtM3z4FzLNlwtEpKyGGtyl2k7OehNCd
CQf2IcW+GV9gdn5n1jFYPsE21h/nCuHtt38n6sXk1lhBGv7t6qI18Rf13hnMg1Fq0qCZjUMO6rXzfESetBO6N
L1zF/wCIaM0r7oilummeK6gYVWukFwz3W+t2OYnNnJk5nPzPPozI49RRvaNDA==
```

This string allows the Policy Validator to identify the users. This identity can then be used to present customized content for that user, if the administrator activates personalization attributes that correspond to the user's attributes.

For details on how to use activated attributes to customize content, see *Building the Users Tree* on page 15 in the *Policy Builder User's Guide*.

In most cases, the cookie prevents users from having to reauthenticate each time they request a new Web page on that server. However, depending on the type of cookie users have, they might be required to reauthenticate between sessions:

- Single-session nonces help the Policy Validator keep track of a user during a single session.
- Inter-session nonces are persistent and have an expiry date, after which time the Policy Validator again requires users to authenticate themselves.

How Select Access uses nonces and cookies to authenticate users

Select Access uses nonces and cookies to authenticate returning users through a simple process:

1. The Policy Validator includes a nonce in the XML reply it sends to the Enforcer plugin.
2. The Enforcer plugin sets the value of the nonce as a cookie and sends it to the HTTP client application.
3. The client application passes the cookie back to the Web server in later requests in order to inform the Policy Validator that the user has already been authenticated.
4. The Policy Validator may generate a new nonce that includes a new timestamp, and the cycle begins again.
5. If the user ever fails to make a new request within the time limit encoded in the nonce's timestamp, the Policy Validator acts as it did on the first request; that is, it ignores the nonce and takes the

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

unauthenticated path through a conditional access rule if one exists.

6. If the result of evaluating the conditional rule is deny, the reply sent to the client application may include information indicating that access might have been permitted had authentication taken place.

Nonces are ignored during evaluation of conditional rules that do not contain an authentication decision point, and do not use SelectID.

Forcing user logout

By manipulating the way cookies are created and used, administrators can force users to explicitly log out from your Web site. When users log in to a Web site protected by Select Access, they are issued an HTTP cookie representing their logon session. This cookie typically expires after ten minutes of idle time—a value you can configure (for details, see *To define the Policy Validator's data encryption settings* on page 132 of the *HP OpenView Select Access 6.0 Installation Guide*). When the cookie expires, the user must log on again if he wishes to access controlled content. The lifetime of the Select Access session cookie also ends when the user completely closes the Web browser. When Select Access receives a request with a cookie nearing its expiry time, it issues a refreshed cookie indicating that the session is not idle.

For some applications, particularly those where users are using shared machines, it is desirable to allow users to explicitly log out so that subsequent users of the shared machine cannot take advantage of the session cookie. There are two ways to avoid this:

- Request that the user close all browser windows on the machine; however, this is difficult to enforce.
- Log out the user with an invalidated session cookie. To log out users, Policy Validator replaces an existing valid HTTP session cookie with an invalid cookie. Invalid cookies cause future connection attempts to fail during the authentication. When Select Access receives an invalid cookie from the Web browser, it forces users to log in again in order to see controlled resources.

Setting up Web content for logout

To use the logout feature with your Web content:

1. Create a new rule. For details, see *Creating a rule* on page 121 of the *Policy Builder User's Guide*.
2. Add a **Logout User** terminal point.
3. Save this rule.
4. Select the user and network resource pair and apply the corresponding conditional access control rule to it in the Policy Matrix.

5. Any time a user follows a link to this page, the Select Access logout rule is invoked and the user's cookie is replaced with the invalid cookie.

We recommend that you make the logout URL a separate page, rather than display it within a frame. The reason for this is that some Web browsers make parallel requests for the contents of each frame. If two or more of those frames contain URLs protected by Select Access, it is remotely possible that an updated cookie is sent in response to one URL while the logout cookie is sent in response to the other. In this case, which cookie the Web server stores cannot be predicted. This problem does not occur if the logout URL is not included in a frameset.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

Chapter 9

Understanding certificate-based authentication

Select Access supports certificate-based authentication for all its communications—whether it is between users and servers or whether it is among Select Access components. By forcing users and components to authenticate themselves with a trusted certificate, Select Access creates a more trusted environment.

Processing a certificate query can be very time-consuming. It is an operation that involves multiple processes. Occasionally a certificate query for a transient user might take priority over another query. If another query is interrupted by a certificate query, an informational message is logged that says: `short-circuiting`. Do not be alarmed by this message; it means that the Policy Validator is not rerunning the complete certificate verification process.

Authenticating components with certificates

Select Access components communicate over TCP/IP, a protocol that enables two hosts to establish a connection and exchange streams of data. Because a single machine can host multiple instances of Select Access components, it is important that connections between Select Access components be protected.

If your default directory server (the one you define at the start of the Administration server's setup) runs over SSL, Select Access must be configured to use certificates. There are two ways you can choose to set up certificate-based authentication among components, to ensure these data streams are well protected:

- **Let Select Access handle certificates**: Select Access can distribute, manage, and verify components' certificates required for the negotiations of SSL connections. In this instance, configuration of certificate-based authentication and data encryption requires no intervention and additional setup is not required.
- **Import your own CA certificate**: You can import your own external CA certificate. The Administration server uses this certificate to authenticate and negotiate SSL connections. In this instance,

intervention is required to configure authentication and data encryption, causing the setup to become somewhat more complex.

The Administration server and certificate management

SSL certificates for your Select Access components are managed by the Administration server. As part of its duties, the Administration server is entirely responsible for the management of the CA certificate as well as the distribution of newly signed certificates to Select Access components.

There are two sides to certificate management:

- How the Administration server manages the certificates it creates
- How Select Access components validate certificates

The topics are described in the sections that follow.

How the Administration server manages certificates

Immediately after its installation and configuration, the Administration server waits for subsequent certificate requests from the InstallAnywhere wizard. This wizard acts on behalf of the administrator (that is, the installer uses the administrator's credentials). It verifies the request before creating a signed certificate using a CA certificate stored on the directory server (which you may or may not have imported), and registers the component that has been issued a signed certificate thereby building a list of trusted components.

Carefully maintain this list of trusted components so that the Administration server can effectively manage certificates of those components it believes to be trusted. That means that if there is a perceived security transgression of any kind—especially one involving certificates or keys—you need to uninstall the affected component and remove it from any component list it belongs to (for example, the list of available Policy Validators used by Enforcer plugins).

When trust is breached

You would likely want to remove a trusted component when you need to address a possible security transgression involving one or more components.

Regenerating SSL certificates

Regenerating your SSL certificates synchronizes them despite any change in your deployment. Regenerate certificates anytime there are concerns as to whether or not a change in your system would affect the way certificates are processed among all Select Access

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

components. Important deployment changes that require SSL certificate regeneration include:

- When you remove a component whose trust has been breached. You cannot just stop a Policy Validator, or copy or delete files required by an Enforcer plugin; otherwise the list of trusted components is not adequately maintained by the Administration server, which can degrade the security of the Select Access system.

  To update the component list (stored on the directory server or IDs used by other components):

  a. Uninstall the offending component from the host machine with the installer.

  b. Reinstall the component in question. This creates a new ID for it as well as regenerates its SSL client certificate for it.
- If you move or reinstall the Administration server. You can only have one Administration server on your Select Access-protected network. If you reinstall the Administration server elsewhere, regenerate certificates for all the components on your network.
- If your CA changes. In this case, you would need to regenerate certificates for all Select Access components.

How components process certificates

The SSL system that Select Access employs creates three levels of validation. Components check that:

1. The certificate presented is a valid certificate, because it comes from a trusted source.
2. The certificate confirms that the requesting component is what it claims to be. All certificates place the component's host name or IP address into the certificate.

HP recommends that you use a host name in the certificate rather than an IP address as hostnames are less likely to change. If you change a component's IP address, the certificate for that component is no longer valid. You need to rerun the installer on the host machine so that you can regenerate a certificate to that machine that matches its changed IP address.

3. The component is allowed to connect and open an SSL-encrypted session. Using the component's ID in the certificate, the name or IP address is compared against the list of registered components.

This method of mutual authentication and subsequent data encryption decreases the likelihood of unauthorized access to the data stream as well as the data source.

Authenticating users with certificates

There are three categories of users Select Access can authenticate:

- The administrator who signs policy data to prevent indirect tampering with it. The data signer's certificate is configured with the Administration server. You can either let Select Access handle the managment of this certificate, or import your own.
- The local or remote administrator who runs the Policy Builder applet over HTTPS. The administrator's certificate is configured with the Administration server. You can either let Select Access handle the managment of this certificate, or import your own.
- Known and unknown users who need to be authenticated via a certificate authentication server. Once have been authenticated, users can open an encrypted SSL connection using the HTTPS protocol.

This section describes the latter kind of user authentication.

How Policy Validator performs user lookups

When the Policy Validator begins analyzing the user's certificate to determine its validity, as shown in Table 34.

Table 34: Policy Validator lookup process

This step...	For details, see...
1. Find the user to see that the individual making the request can be authenticated.	*Finding the user* on page 135
2. Verify the certificate the user is presenting is a valid certificate.	*Verifying the validity of the certificate* on page 136
3. Create a transient user entry so the Policy Validator can authenticate future access requests	*Creating a transient user* on page 136

Only after these steps take place does the policy for that user get checked and a final allow or deny decision is made. If the Policy Validator authorizes access to an HTTPS resource, the user's certificate is checked and the session is then encrypted with the algorithm in this certificate. Otherwise, if the resource is an HTTP resource, the certificate is ignored.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

Finding the user

When trying to locate the user, Policy Validator extracts information from the certificate and tries matching an entry against it. The procedure below summarizes this process:

1. The user presents a certificate to the Policy Validator, which then validates it via the certificate server. Often, the `SubjectDN` of the certificate is similar to that of the users User Tree entry on the LDAP directory server.

If you have configured **Advanced Certificate** properties, then certificate CN may be mapped to another user attribute. If this is the case, then the lookup behavior uses the attribute you have mapped to the CN instead, rather than the `SubjectDN`. For details on how you configure **Advanced Certificate** properties, see *To configure advanced certificate properties* on page 102 in the *HP OpenView Select Access 6.0 Policy Builder Guide*.

---

2. The Policy Validator checks its cache to see if a `SubjectDN` exists from a previous lookup.
3. If no cached information matches, the Policy Validator looks for a matching `SubjectDN` on the directory server. One of two methods for locating a user entry is used:
   - If the certificate's `UID` is the user's `RDN` (assuming all `UID`s are unique), the Policy Validator extracts the `UID` and looks in LDAP for the user's authorization policy.
   - If not, the Policy Validator does a `CN` search after it extracts the `CN` for the `SubjectDN`. The Policy Validator only checks the policy if the user entry contains a compatible certificate.

   The first match is automatically returned. For example, say you have the following user entries:

   ```
   CN=Jane Smith, OU=Sales, O=mycompany.com, C=CA
   CN=Jane Smith, OU=Development, O=mycompany.com, C=UK
   ```

   Therefore if the Policy Validator starts looking for "CN=Jane Smith" the Policy Validator finds both entries. However, it does not know which one matches the certificate, so it looks for that certificate in the user's entry on the directory server.
4. If no match occurs, then a transient user entry is created, and an entry is added to the Users Tree.

Create this transient user location before configuring the certificate server.

Verifying the validity of the certificate

Verifying the validity of a certificate is a two-step process that checks to see if:

1. The user's certificate is a compatible certificate.
2. The certificate is valid and has not been invalidated. The Policy Validator does this by looking up the certificate's revocation status with:
   - CRL (Certificate Revocation Lists)
   - OCSP (Online Certificate Status Provider)

Creating a transient user

The Policy Validator creates a transient user when:

- The user entry is not in LDAP, but in the Certificate server's database.
- The certificate provided is a valid certificate.

This allows the Policy Validator to look up the user's policy on the directory server, even though no user entry exists for that user. It does this by concatenating the certificate's `UID` or `CN`, and "synthesizes" user entries by permanently caching data. For details, see *Validating users when their entries are not on the directory server* on page 79 in the *HP OpenView Select Access 6.0 Policy Builder Guide*.

Understanding cryptographic algorithms used by certificates

Cryptography uses algorithms to accomplish the following:

- Keep communications private.
- Protect sensitive information.
- Identify and authenticate the parties involved in the communications.

Although several algorithms exist, Select Access uses the ones that are fast, easy to use, and that have good security properties. Select Access uses these specific algorithms for the following functions:

- Nonces
- Certain aspects of SSL
- Certain aspects of policy signing

Encryption methods

The following encryption methods are the most common, and are often used with an authentication method such as digital signatures to vouch for a message's integrity.

- Symmetric key cryptography (also known as private key cryptography). The algorithms in this category have the following characteristics:

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

  - They use only one key for both encryption and decryption.
  - The security of this type of algorithm rests in the key, therefore it must remain secret or *private*.
- Asymmetric key cryptography (also known as public key cryptography). The algorithms in this category have the following characteristics:
  - The key used to encrypt information is different from the key used to decrypt information.
  - The key used to encrypt, called the *public* key, can be made public without compromising the message's security.
  - The key used to decrypt, called the *private* key, is held by a specific person.
- Cryptographic message digest. This type of algorithm allows you to output a hash or digest that you compare to a digest generated locally to check, for example, that your email or certificate has not been tampered with.

Table 35 provides an overview to how some common algorithms can be used by Select Access.

Table 35: Common algorithms used by Select Access

Algorithm	How used
MD5 (also known as Digest or Message Digest)	Used to create secure user credentials such as cookies and nonces. For details, see *Understanding nonces and cookies* on page 126. **Caution!** Using this algorithm generates the cookie at a faster rate, but anyone with access to the `validatorData` object in the directory server can forge cookies.
RSA (Rivest, Shamir, and Adleman)	Used for both encryption (opening SSL sessions) and authentication. For example, if selected during the Policy Validator's configuration, can be used to create secure user credentials such as cookies and nonces. **Note:** With this algorithm, you get the highest level of security, but the cookie is generated and verified at a slower rate.
SHA-1 (Secure Hash Algorithm)	Used to create digital signatures to secure policy data, because it produces a 160-bit hash value.

Table 35: Common algorithms used by Select Access (Continued)

Algorithm	How used
3-DES (Triple Data Encryption Standard)	If you import a CA certificate that uses this symmetric algorithm, you can use it to open SSL sessions.
AES (Advanced Encryption Standard)	If you import a CA certificate that uses this symmetric algorithm, you can use it to open SSL sessions.
RC4 (Rivest Cipher)	If you import a CA certificate that uses this symmetric algorithm, you can use it to open SSL sessions. This algorithm is three times faster than DES'.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

Appendix A

Integrating the servlet Enforcer plugin with a servlet engine

Integrating the servlet Enforcer plugin on a servlet engine requires that you configure both technologies, specific properties and parameters to ensure they function properly as a unit. For details, see the corresponding section below:

- *Integrating the servlet Enforcer plugin with a servlet engine* on page 140
- *Configuring Select Access with Tomcat and the servlet Enforcer plugin* on page 143
- *Testing your deployment* on page 145

If you require details on how to integrate with WebSphere and WebLogic, check the respective integration papers.

In order to use the servlet Enforcer plugin, you must first download the JCE extension from Sun's Web site. Due to legal restrictions, HP cannot ship this extension with Select Access.

To download this extension, go to:

```
http://java.sun.com/products/jce/index-122.html.
```

After you unpack the zipfile, you'll find jar files in the lib directory. You'll need to add `jce1_2_2.jar`, `US_export_policy.jar` and `local_policy.jar` to your `CLASSPATH`. You should then be able to run under the JRE1.3.

Integrating the servlet Enforcer plugin with a servlet engine

This section documents how to integrate the servlet Enforcer plugin with a servlet engine, using Tomcat as an example application server. When you integrate Tomcat with the servlet Enforcer plugin, you are configuring it to run with the servlet Enforcer plugin. Table 36 outlines the high-level setup tasks and their corresponding implementational steps that you must consider.

Table 36: Integrating the servlet Enforcer plugin with Tomcat—procedure summary

This setup task...	Steps to accomplish it...
Step 1: Copy Select Access JAR files into Tomcat's common folder. Copying these files allows Tomcat to put these files into its classpath upon startup.	1. Locate the following files from either the `<Select_Access_install_path>/shared` (for unsigned JARs) or the `<Select_Access_install_path>/jetty/protected/` (for signed JARs) folders: – `castor-0.9.3.19-xml.jar` – `EnforcerAPI.jar` – `jakarta-oro-2_0.jar` – `jce.jar` – `jcert.jar` – `jcrypto.jar` – `jdom.jar` – `jpkiplus.jar` – `jssl.jar` – `ldapjdk.jar` – `pkcs11.jar` – `protomatter.jar` – `SAResourceBundle.jar` – `servletenforcer.jar` – `shared.jar` – `xerces.jar` – `xml.jar` 2. Copy all of these jar files to the following directory: `<Tomcat_install_path>/common/lib`.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

Table 36: Integrating the servlet Enforcer plugin with Tomcat—procedure summary (Continued)

This setup task...	Steps to accomplish it...
Step 2: Modify the Web application's deployment descriptor file called `web.xml`. Use this file to define: • Where the servlet Enforcer plugin can find its files. • 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 Web application, and its resources, can not be Enforcer-protected. **Note:** Tomcat's administration tool does not allow you to change the application's configuration file to use servlet filters. You must make this change manually.	1. Create a new `<filter>` section, and define the servlet Enforcer plugin as one of your Web application's filters. If the application requires more than one filter, ensure the servlet Enforcer plugin is loaded as a filter first. 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. **Note:** The value of the servlet Enforcer plugin's configuration file's parameter is only the name of the file. The global configuration file contains a parameter (that is, `SELECTACCESS_CONFIGS`) that 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. Code example 14 is an example of a `web.xml` file that has been modified to include these Select Access-specific modifications.

Code example 14: Example web.xml file

```
<?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>
```

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

Configuring Select Access with Tomcat and the servlet Enforcer plugin

Table 37 outlines the high-level setup tasks and their corresponding implementational steps that you must consider when setting up Select Access to run with the Tomcat servlet engine.

Table 37: Setting up Select Access

This step...	Details on how to do it...
Step 1: With the Setup Tool create a bootstrap XML file that contains configuration parameters required by the servlet Enforcer plugin. Currently the servlet Enforcer plugin must use the Generic Enforcer plugin wizard in the Setup Tool.	1. Run the Setup Tool and click **Next** until the Generic Enforcer plugin setup wizard appears. 2. In the **Choose the location** field, enter the full path and a filename for file that will be created. Using Tomcat as the example, enter the following: `<`*`Tomcat_install_path`*`>/common/lib/<`*`enforcer_name`*`>` 3. Click **Configure** to begin the configuration. 4. On the **General** setup screen choose a **Custom** setup. 5. On the **ID** setup screen, enter a specific name for the servlet Enforcer plugin used by Tomcat. This ensures that the ID in the XML file and the ID on the directory server match. 6. Modify the remaining screens as necessary. **Note:** While you can modify the remaining entries if you choose, the Select Access defaults are acceptable for the servlet Enforcer plugin. 7. On the **Finish** setup screen, ensure both check boxes are unchecked before clicking the **Finish** button. This creates a file called `<`*`enforcer_name`*`>.xml` in the `<`*`Tomcat_install_path`*`>/common/lib` folder. For additional details, see Chapter 8, *Configuring the Enforcer plugins* in the *HP OpenView Select Access 6.0 Installation Guide.*
Step 2: Add the Tomcat servlet engine to the Resources Tree as a service branch.	1. Right-click a folder or the root of the Resources Tree. 2. Click **Run Discovery>Services.** The **Discover Networks Services** dialog box appears. 3. Provide the required information on the **Networks** and **Protocols** tabs and then click **OK**. For details, see Chapter 3, *Building your Users and Resources Trees* in the *HP OpenView Select Access 6.0 Policy Builder Guide*.

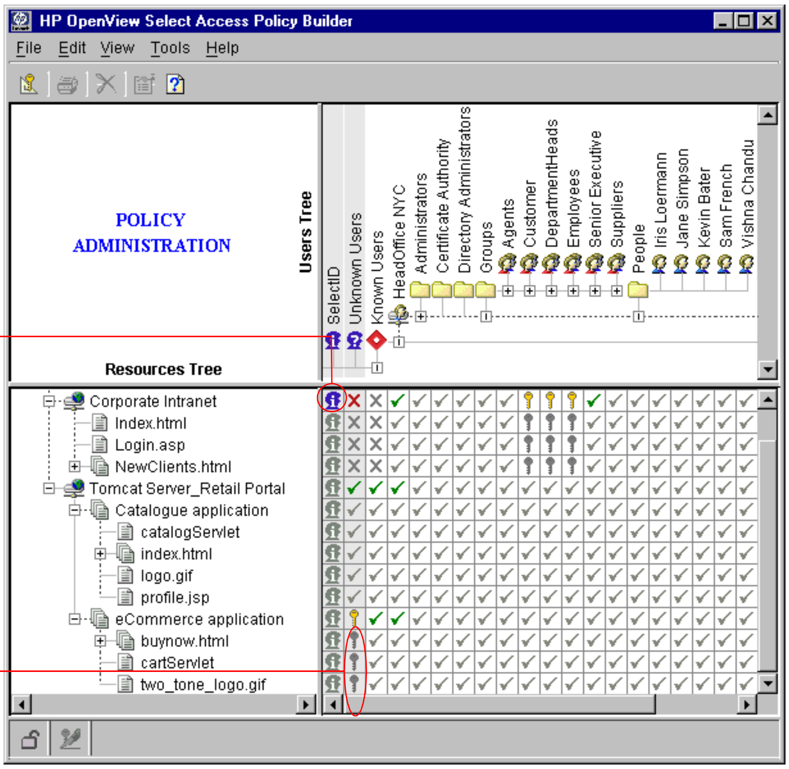
Table 37: Setting up Select Access

This step...	Details on how to do it...
Step 3: On this new service branch, add each Enforcer-protected Web application and its corresponding servlets, HTML pages, JSP scripts, graphics, etc that are part of this application. When you have added all resources to the service branch, it appears similar to the one shown in Figure 23.	1. Make sure you have configured the network resource plugin. For details, see Chapter 3, *Building your Users and Resources Trees*. 2. On the Resources Tree, right-click the service you want to scan for available resources. 3. Click **Run Discovery>Resources**. The **Discover Network Resources** dialog box appears. — Information about the service's representative server is entered automatically in the **Protocol, Hostname**, and **Port Number** fields. (This information is taken from the service's properties.) — If you have configured a plugin for the protocol used by the service, the plugin's configuration details are entered automatically in the **Plugin Settings** field. 4. Select **Run Resource Discovery Plugin** and fill in any empty fields. 5. Select the location on the Resources Tree to add the resources. Do the following: a. Click the **Browse** button beside the **Network Resources Tree Destination** field. The **Select Resource Destination** dialog box appears. b. Select a service, then click **OK**. 6. Click **OK.** For details, see *Automatically generating a list with a discovery plugin* on page 34 in the *HP OpenView Select Access 6.0 Policy Builder Guide*.
Step 4: Customize your Select Access authentication forms and copy them to the content directory you configured in your servlet Enforcer plugin's setup as well as the location named in your Web application's `web.xml` file.	1. 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 users. 2. Ensure that this location matches the location in all Enforcer-protected Web application's `web.xml` file.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

SelectID is enabled against the network using the Registration authentication server. The servlet Enforcer plugin displays the registration form when a user first tries to access any corporate resources.

A conditional rule is set against Unknown users requesting access to the eCommerce application. Unknown users can only access these resources if they register with the company first.


HP OpenView Select Access Policy Builder
File Edit View Tools Help
POLICY ADMINISTRATION
Users Tree
Resources Tree

	SelectID	Unknown Users	Known Users	HeadOffice NYC	Administrators	Certificate Authority	Directory Administrators	Groups	Agents	Customer	DepartmentHeads	Employees	Senior Executive	Suppliers	People	Iris Loermann	Jane Simpson	Kevin Bater	Sam French	Vishna Chandu
Corporate Intranet	ⓘ	✗	✗	✓	✓	✓	✓	✓	✓	🔑	🔑	🔑	✓	✓	✓	✓	✓	✓	✓	✓
Index.html	ⓘ	✗	✗	✓	✓	✓	✓	✓	✓	🔑	🔑	🔑	✓	✓	✓	✓	✓	✓	✓	✓
Login.asp	ⓘ	✗	✗	✓	✓	✓	✓	✓	✓	🔑	🔑	🔑	✓	✓	✓	✓	✓	✓	✓	✓
NewClients.html	ⓘ	✗	✗	✓	✓	✓	✓	✓	✓	🔑	🔑	🔑	✓	✓	✓	✓	✓	✓	✓	✓
Tomcat Server_Retail Portal	ⓘ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Catalogue application	ⓘ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
catalogServlet	ⓘ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
index.html	ⓘ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
logo.gif	ⓘ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
profile.jsp	ⓘ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
eCommerce application	ⓘ	🔑	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
buynow.html	ⓘ	🔑	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
cartServlet	ⓘ	🔑	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
two_tone_logo.gif	ⓘ	🔑	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Figure 23: Example Policy Matrix with Tomcat resources

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/Tomcat server combination to ensure that you can:

- Start your Policy Validator.
- Start your Tomcat servlet engine. Do this by running the following script: <*Tomcat_install_path*>`/bin/startup.sh`.

⚠ Ensure you set all scripts in the <*Tomcat_install_path*>`/bin/` directory to executable, by using the following command: chmod + x *.sh

- Access Enforcer-protected HTML and/or JSP. This ensures that:
  - The servlet Enforcer plugin has initialized and is mapped to the URL pattern correctly. 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 user has requested is served by Tomcat. Also check that the Policy Validator is processing incoming requests correctly.

Running the servlet and Axis Enforcer plugins together

If you are hosting Axis in a Tomcat servlet container, and have installed both the Axis Enforcer plugin and the servlet Enforcer plugin, you will need to perform a couple of tasks in order to ensure that both function properly:

- Axis files must be added to the servlet Enforcer plugin's list of ignored filenames. See *To create a list of ignored filenames* on page 157 of the *HP OpenView Select Access 6.0 Installation Guide*.
- Delete the following jars from the `axis/web-inf/lib` directory except the following:
  - `castor-0.9.3.19-xml.jar`
  - `EnforcerAPI.jar`
  - `jakarta-oro-2_0.jar`
  - `jdom.jar`
  - `ldapjdk.jar`
  - `protomatter.jar`
  - `SAResourceBundle.jar`
  - `shared.jar`
  - `xerces.jar`
  - `xml.jar`

These files must be deleted, otherwise the Tomcat class loaders will load them twice—once when loading the jars deployed with the servlet Enforcer plugin, and again when loading the jars deployed with the Axis Enforcer plugin. On loading, classes that use JNI will attempt to load some dynamic libs. However, due to a JVM restriction,

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

these libraries cannot be loaded twice and the classes will therefore fail to load.

Additional servlet Enforcer plugin configuration notes

To additional configuration notes for the servlet Enforcer plugin:

- In order to get POST-through-auth working with the servlet Enforcer plugin, you must add the following setting to the Custom Settings setup in the Setup Tool:

  ```
  EXPORT_POST_DATA
  ```

- In order for virtual servers to function properly and reliably, you must add an additional parameter in your `web.xml`:

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

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

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

Appendix A

Integrating the WSE Enforcer plugin into the .NET Framework

Although the Select Access installer and Setup Tool take care of installing and configuring the WSE Enforcer plugin, there are a few things you will need to know in order to fully integrate the plugin into your .NET Framework. For details, see the corresponding section below:

- *Prerequisites for installing and running the WSE Enforcer plugin* on page 149
- *Configuring the WSE Enforcer plugin* on page 150
- *About SOAP signing and encryption using the WSE Enforcer plugin* on page 152
- *Using ASP .NET process identity on Windows 2000 and Windows 2003 Server* on page 154
- *Disabling HTTP-GET and HTTP-POST on the web service* on page 154
- *How the Select Access WSE Enforcer plugin works* on page 155

Prerequisites for installing and running the WSE Enforcer plugin

In order for you to install and run the WSE Enforcer plugin, you must ensure that several system requirements have been met.

Before installing the WSE Enforcer plugin

Before the WSE Enforcer plugin can be installed, you must ensure that you have the following components on your machine:

- the Microsoft .NET Framework
- the Web Services Enhancements 1.0 add-on

During the installation process, the Select Access installer searches the installation machine to determine whether the prerequisite files are present. If it does not find them, it makes the WSE Enforcer plugin component unavailable for installation.

HP recommends testing your environment by ensuring that you are able to access your web service using a browser. If you have already enabled WSE and SOAP extensions for your web service, then it would also be helpful to invoke methods on your web service and access it using a SOAP client.

The administrator must have an understanding of how credentials are passed in SOAP messages using WSE. For more information on sending security credentials in SOAP message, see the SOAP documentation.

For information on installing these components, refer to your .NET documentation.

Before running the WSE Enforcer plugin

In order for the WSE Enforcer plugin to function, the following two assemblies, must be added to the .NET Framework General Assembly Cache (GAC):

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

When you select this plugin for installation, the Select Access installer searches for the GAC file (`gcutil.exe`), which it uses to automatically install the necessary assemblies during the installation process.

However, if the installer is unable to locate the file, then the required assemblies will not be installed automatically. In this case, you must manually add the assemblies to the GAC.

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 on 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**.
   a. Select **Action>Add**. The **Add an Assembly** dialog box appears.
   b. Select the `WSEEnforcer.dll` and click **Open**.
4. Close the **.Net Framework Configuration** window.

Configuring the WSE Enforcer plugin

The WSE Enforcer plugin is fully configurable from the both the Setup Tool and via the Policy Builder's **Component Configuration** window. For more information on using these utilities to configure the plugin, see:

- *Configuring the Enforcer plugin* on page 139 of the *HP OpenView Select Access 6.0 Installation Guide*

- *Modifying group and override parameters for the Enforcer plugin* on page 228

How the setup tool alters the web.config files

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 of the selected Web Services to place the WSE Enforcer plugin in the sequence of SOAP filters.

The Select Access input and output filters and the password provider are added to the configuration file.

If another password provider is already configured in the `web.config` file, the Setup Tool will not be able to update the file and will display a warning message.

If signing is enabled and a certificate store is configured for signing, the Setup Tool will update the `storeLocation` attribute of the `<X509>` element with the chosen certificate store. The certificate store will also be added to the enforcer XML configuration that is saved to the directory server.

If the certificate store configured for signing is modified using the Component Configuration tool in the Policy Builder, only the WSE Enforcer plugin configuration file will be updated with the new store. The `storeLocation` attribute of the `<X509>` element in the `web.config` files of the protected web services will not be updated, since the Component Configuration tool does not access this file.

If you want to change the certificate store location, you should always use the Setup Tool to change this setting (found in the XML SOAP Signing setup screen), not the Component Configuration tool of the Policy Builder.

Code example 15 shows the segments of XML that the Setup Tool adds to the `web.config` files:

Code example 15: Select Access input and output filters added by the Setup Tool

```
<configuration>

<system.web>
   <webServices>
   <soapExtensionTypes>
   <add type="Microsoft.Web.Services.WebServicesExtension, Microsoft.Web.Services,
Version=1.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" priority="1"
group="0" />
   </soapExtensionTypes>
   </webServices>
</system.web>
```

```
<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>
   <security>
   <passwordProvider type="com.hp.ov.selectaccess.enforcer.wse.PasswordProvider,
WSEEnforcer, PublicKeyToken=..., Culture=neutral, Version=1.0.0.0" />
   <x509 storeLocation="LocalMachine" />
   </security>
</microsoft.web.services>
```

To understand the configuration settings in the `web.config` file, refer to the WSE 1.0 documentation.

Unprotecting Web Services

If the administrator chooses to unprotect protect web services (by deselecting them from the protected web services list on the Select Web Services dialog available via the access policy), then the Select Access WSE input and output filter will be removed from the `web.config` configuration files of the affected services. The remaining XML in the file is untouched.

About SOAP signing and encryption using the WSE Enforcer plugin

You can secure SOAP messages sent by web services by configuring the WSE Enforcer plugin to sign and encrypt all SOAP messages.

To read more about SOAP signing and encryption, refer to the WSE documentation.

Signing SOAP messages

You can sign outgoing SOAP messages with the WSE Enforcer plugin using one of two methods:

- Certificate: The WSE Enforcer plugin will attempt to retrieve the certificate from the configured certificate store and the private key from a folder on disk. For the WSE Enforcer plugin to access these files, the administrator must install the certificate store and ensure that the correct permission is set on the folder containing the private key or on the private key file itself.

  Private keys are stored in the folder:

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

- Password: The WSE Enforcer pluginl creates a Username Token comprised of the values you specify for the username, password and password option parameters. Three password options are available:
  - SendPlainText
  - SendHashedText
  - SendNone

  If you choose a password option other than SendPlainText, it is expected that the receiver of the SOAP response message has the password available and can use that to validate the sent Username Token.

If you are using a password to sign SOAP messages sent to web services that are protected by Select Access, you must select the SendPlainText format for the password.

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. For security purpose it is recommended that the web service run over SSL.

For more about Username Tokens and Password Options, see the WSE documentation.

When you enable SOAP signing for outgoing responses, the WSE Enforcer plugin will sign the SOAP Body and `<Policy>` statements in the SOAP header.

You can also choose to require all incoming SOAP messages to be signed. When you select this option, the WSE Enforcer plugin checks that the SOAP Body of the request message was signed.

For information on setting SOAP signing options, see *Setting up SOAP message signing* on page 152 of the *HP OpenView Select Access 6.0 Installation Guide*.

Encrypting a SOAP Message

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 signature.

For information on setting the SOAP encryption option, see *Setting up SOAP message encrypting* on page 154 of the *HP OpenView Select Access 6.0 Installation Guide*.

Troubleshooting certificate errors

Should you experience difficulty using certificates with SOAP, the WSE documentation

Using ASP .NET process identity on Windows 2000 and Windows 2003 Server

If you intend to use the WSE Enforcer plugin to protect Web services, it is important that you understand the identity under which ASP. NET runs on your machine so that you can give the appropriate accounts access to the private keys. The identity of an ASP. NET Web Service is configured in the `<processModel>` element of the following file:

```
<.NET_inatation_dir>\Framework\<version>\CONFIG\machine.config
```

Different accounts are used depending on your platform/IIS configuration.

- Windows 2000: By default, ASP .NET runs as `ASPNET local`.
- Windows 2003: When ASP. NET is running under IIS 6 in worker process 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. The default process identity in this case is `Network Service`.

  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.

Disabling HTTP-GET and HTTP-POST on the web service

Web services can be configured so that they are invoked by any of the following methods:

- HTTP GET
- HTTP POST
- HTTP POST using SOAP

The WSE Enforcer plugin filters are configured exclusively for SOAP requests, and therefore can only be invoked by HTTP POST requests that contain SOAP messages. They cannot be invoked with HTTP GET or HTTP POST requests that do not contain SOAP.

HP recommends that you disable the unused methods in the web.config files of every protected web service. HTTP GET and HTTP POST can be disabled by adding the following XML to the `web.config`.

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

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

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

Microsoft recommends that rather than disabling HTTP-GET and HTTP-POST for each service, you should disable these methods on the entire machine and only enable HTTP-POST via SOAP to secure web services. For more information, see Microsoft's SOAP documentation.

How the Select Access WSE Enforcer plugin works

On receiving an incoming SOAP request, the WSE Enforcer plugin acts as an input filter and constructs a Policy Validator query from the request data. The service URL specified in the query is the URL from which the web service was accessed. The resource name will be obtained 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.

The Select Access WSE Input filter will then extract any credentials (the username/password or certificate) sent in the SOAP request. If more than one set of credentials is received in the SOAP request, the first found will be used, and all others ignored. Once the credentials are extracted and added, the WSE Enforcer plugin sends the query to the Policy Validator.

The WSE Enforcer plugin can extract credentials sent in a SOAP request from a Username Token and a Binary X509 Security token. Below is the list of Select Access authentication servers supported for .Net Web Services and the security token which can be used to store the security credentials:

- Password - credentials sent in Username Token
- Certificate - credentials sent in X509 Binary Security Certificate Token
- SecurID (next pin scenario is not supported) - credentials sent in Username Token
- Windows Kerberos - credentials sent in Username Token with the domain name must be prepended to the username
- Windows NTLM - credentials sent in Username Token with the domain name must be prepended to the username

For 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.

When the Policy Validator allows the request

If the Policy Validator returns an `ALLOW`, the WSE Enforcer plugin will extract all cookies from the reply and send them on to the web service client. All of the Policy Validator cookies will be added to the HTTP response header, but only the `PolicyUser` cookie will be added to the SOAP header and returned in the SOAP response message. Code example 15 illustrates what the format of the SOAP XML will be.

Code example 16: Format of 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

When an authentication failure occurs, 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 must inform the client so that it can send the correct credentials in the next request. The policy assertions contained within a policy statement can be used to express, among other things, the authentication requirements of the Web Service. WS-Policy specifies the XML grammar of the policy statement.

If no credentials or invalid credentials were sent, the Policy Validator replies with a `DENY`. The reply may also contain the type of authentication server needed for successful authentication. shows a sample deny response, with additional authentication information supplied.

Code example 17: 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>
```

In order to give the client more information about the kind of authentication required to access a target method/resource on the web service, besides adding the SecurityToken element in Policy statement, the WSE Enforcer plugin also sends additional information about the type of authentication method the client is expected to authenticate with. For every authentication method returned in the

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

Policy Validator reply, a new `<AuthType>` element belonging in the Select Access namespace will be added under `<Policy>`. Code example 18 illustrates the `<Policy>` XML that results from the above Policy Validator reply.

Code example 18: <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>
<sawss:AuthType xmlns:sawss="http://www.hp.com/SelectAccess/webservices/security">
password</sawss:AuthType>
</wsp:Policy>
...
</soap:Header>
```

In addition to the `<Policy>` statement, the WSE Enforcer plugin also returns a SOAP fault in the SOAP response message. The WSE Enforcer plugin returns the following fault codes:

- `Client.FailedAuthentication`—The client did not sent valid credentials and could not be authenticated
- `Server.UnableToAuthenticate`—WSE Enforcer 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`—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

The web service client on recognizing the fault should look up the Policy statement and attempt to send the correct credentials in the next SOAP request.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

Appendix B

Monitoring Select Access operations

Alerting is a feature that notifies an administrator via email of significant incidents involving users and components.

Alerts can be configured in the following two places:

- The Rule Builder's alert decision point: This component sends an alert notification wherever the audit policy defines such an alert should go, when a user reaches this point in a rule. For example, an administrator could insert an alert decision point in a rule that he applies against each of his company's confidential resources, such as payroll information, employee records, and prospective new customers. For details, see *Alert notification decision point* on page 159.
- The Secure Audit server's **Audit Trail** tab: This component alerts one or more email addresses when preset conditions are met. Event notification by email is similar to other logging destinations. Because the number of email messages generated by a Select Access component can be overwhelming, use this feature primarily for serious component issues. For example, if a runtime component fails or an unexpected error occurs in the system, an email notifies the administrator of the problem. For details, see *Event notification via email* on page 160.

While the alert notification decision point involves monitoring users' movements, event notification via email involves monitoring components' activities.

Types of alerts

The two types of alerts described below can be configured to notify an administrator immediately of an incident, regardless of whether it involves users or components.

Alert notification decision point

The alert notification decision point triggers a log message—which can be routed using the Policy Validator's audit settings—when a user passes or fails a specific criteria in the rule.

The alert notification decision point has the following benefits:

- The administrator gets notified immediately when a user tries to access a resource on which the alert decision point is applied.
- You can filter messages based on their level of severity.

For example, if you want to be alerted each time a user tries to access the company payroll information, even though she is allowed access, you would create a rule that includes this decision point. For details, see Chapter 8, *Creating policy rules with the Rule Builder*, in the *HP OpenView Select Access 6.0 Policy Builder Guide*.

Event notification via email

An email alert is a message sent to addresses you specify, to notify the addressees of a specific (usually severe) message that has been triggered. For example, you can configure email alerts so that an administrator is immediately notified when a Policy Validator fails.

Event notification via email has the following benefits:

- The administrator gets notified immediately when a component experiences a problem.
- You can configure alerts to be sent to several people.
- The alerts can be collected into a report using the Report Viewer.
- You can find out exactly which component sent the alert.

For details, see Chapter 3, *Changing Audit Settings*, in the *HP OpenView Select Access 6.0 Policy Builder Guide*.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

Appendix C

About Policy Validator queries

Because Select Access is a user-centric, policy-based method of access management, user validation and policy enforcement are twin goals of the security model it provides. Select Access attains these goals with two server components: the Policy Validator (the logic engine) and the Enforcer plugin (the Web server agent). Policy Validator is responsible for communicating with the Enforcer plugin via an entity known as a query.

Table 38 outlines major topics covered by this appendix.

Table 38: Query-related topics

Topic...	For details, see...
An introduction to the process by which an Enforcer plugin sends an XML query to the Policy Validator.	*How is a query used?* on page 148
The structure of XML queries and responses.	*What is the structure of an XML query* on page 148
An overview of the Query utility and how you can use it to evaluate the performance of Select Access components within your environment.	*The Query utility* on page 152

What is a query?

A query is an XML-encoded request the Enforcer plugin makes on behalf of a user for a particular resource on the Enforcer-protected Web server. The Policy Validator then evaluates the user's access policy in the Policy Store to determine if the user is allowed or denied access to the network resource. The Policy Validator then returns the access decision to the Enforcer plugin.

How is a query used?

The process by which the Policy Validator and the Enforcer plugin evaluate a user access request is summarized below:

1. The Enforcer plugin gathers information about a user's incoming access request and packages data it discovers as an XML query to the Policy Validator.
2. When a query is received, the Policy Validator deciphers the query and determines where additional data lookups need to be performed for the user and resource pair in question.
3. The Policy Validator accesses the required policy information for the user and resource pair from the Policy Store and interprets the policy logic and conditions surrounding the request. This information becomes cached if it is required in the future.
4. If it discovers a conditional policy, the Policy Validator checks and evaluates the rule criteria defined within one or more conditional rules. A decision is reached.
5. The Policy Validator communicates the result of the policy logic to the Enforcer plugin making the request, which then enforces the policy decision.

For additional details on this process, as well as a graphical representation of these steps, see *How does the Policy Builder work?* on page 7 of the *HP OpenView Select Access 6.0 Policy Builder Guide*.

What is the structure of an XML query

The Policy Validator receives queries from various Enforcer plugins as XML documents. The Enforcer plugin also encodes queries to the Policy Validator using XML. XML allows for complete extensibility with respect to modifying information in the query structure: you can arbitrarily add new attributes with values to a query.

Select Access ignores any attributes for which it has no use.

Thus, as with the authorization rules, any type of data can be passed to the Policy Validator. This can include:

- Binary objects like X.509 digital certificates
- Complex objects like complete emails
- Simple objects like standard text

For example, an XML query from the Enforcer plugin on behalf of Lance Mountain is shown in Code example 13. This query shows the that the Enforcer plugin's **Query Details** parameter has been set to maximal. Additionally, it shows Lance's personalization details forwarded by the SAML partner, as well as a list of information that describes Steve Smith in the directory server. Notice how data is delimited by the following tag pairs:

- `<PolicyValidatorQuery>` and `</PolicyValidatorQuery>` delimit the entire query and all data contained in it.
- `<PROPERTY NAME="`*`name`*`"></PROPERTY>` delimit the value for the property with `"`*`name`*`"`.
- `<PROPERTYLIST NAME="`*`name`*`">` delimit the value for the property list (in this case a nested property list) with *`"name"`*.

Code example 19: Example Enforcer plugin query to the Policy Validator

```
<PolicyValidatorQuery>
<PROPERTY NAME="service">http://mymenu.foodcompany.com:8070</PROPERTY>
  <PROPERTY NAME="path">/test/auth/submit.cgi</PROPERTY>
  <PROPERTY NAME="dstIP">10.10.10.30</PROPERTY>
  <PROPERTY NAME="srcIP">10.10.10.110</PROPERTY>
  <PROPERTY NAME="dstPort">8070</PROPERTY>
  <PROPERTY NAME="srcPort">4001</PROPERTY>
  <PROPERTY NAME="dstHost">mymenu.foodcompany.com</PROPERTY>
  <PROPERTY NAME="protocol">http</PROPERTY>
  <PROPERTY NAME="srcHost">user_isp.com</PROPERTY>
  <PROPERTY NAME="nonce">
AgAAAAAAPl5SEQAAAAA+XlRqc29sMDAxLmNhLmJhbHRpbW9yZS5jb206OTk5OABwYXNzAGNuPXN0ZXZlIGtvd
HNvcG91bG9zLG91PXN0ZXZlLGRjPWNhLGRjPWJhbHRpbW9yZSxkYz1jb20AAGp790LYZinTvdZ8UhpWv4CfkX
tmu8885S0AeHA3vX7qrF353ASKBJ5RS2bJz1qCJXGfzK4vQqfVTT0mcE8yIgJQefoFX+GnVV092WaFYtiOe7Q
jfpSqXtaQmShZC1QlRnAYJVwo5Gx5s4B/THU5BgPKZyvUEJnK/pcsb2hOqCOd</PROPERTY>
  <PROPERTYLIST NAME="post_data_list">
    <PROPERTY NAME="fullname">Lance Mountain</PROPERTY>
    <PROPERTY NAME="arrived">10am</PROPERTY>
  </PROPERTYLIST>
  <PROPERTY NAME="native_nonce">enable</PROPERTY>
  <PROPERTY NAME="http_query">hungry=yes&amp;lunch=pizza</PROPERTY>
  <PROPERTYLIST NAME="http_query_list">
    <PROPERTY NAME="hungry">yes</PROPERTY>
    <PROPERTY NAME="lunch">pizza</PROPERTY>
  </PROPERTYLIST>
  <PROPERTY NAME="method">POST</PROPERTY>
  <PROPERTYLIST NAME="http_header_list">
    <PROPERTY NAME="Host">dev01:8070</PROPERTY>
    <PROPERTY NAME="User-Agent">Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US;
rv:1.0.1) Gecko/20020823 Netscape/7.0</PROPERTY>
    <PROPERTY NAME="Accept">
text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,video
/x-mng,image/png,image/jpeg,image/gif;q=0.2,text/css,*/*;q=0.1</PROPERTY>
    <PROPERTY NAME="Accept-Language">en-us, en;q=0.50</PROPERTY>
    <PROPERTY NAME="Accept-Encoding">gzip, deflate, compress;q=0.9</PROPERTY>
    <PROPERTY NAME="Accept-Charset">ISO-8859-1, utf-8;q=0.66, *;q=0.66</PROPERTY>
    <PROPERTY NAME="Keep-Alive">300</PROPERTY>
    <PROPERTY NAME="Connection">keep-alive</PROPERTY>
    <PROPERTY NAME="Referer">http://dev01:8070/test/allow/postfix.html</PROPERTY>
    <PROPERTY NAME="Cookie">
PolicyUser=AgAAAAAAPl5SEQAAAAA+XlRqc29sMDAxLmNhLmJhbHRpbW9yZS5jb206OTk5OABwYXNzAGNuPX
N0ZXZlIGtvdHNvcG91bG9zLG91PXN0ZXZlLGRjPWNhLGRjPWJhbHRpbW9yZSxkYz1jb20AAGp790LYZinTvdZ
8UhpWv4CfkXtmu8885S0AeHA3vX7qrF353ASKBJ5RS2bJz1qCJXGfzK4vQqfVTT0mcE8yIgJQefoFX+GnVV09
2WaFYtiOe7QjfpSqXtaQmShZC1QlRnAYJVwo5Gx5s4B/THU5BgPKZyvUEJnK/pcsb2hOqCOd</PROPERTY>
    <PROPERTY NAME="Content-Type">application/x-www-form-urlencoded</PROPERTY>
    <PROPERTY NAME="Content-Length">54</PROPERTY>
  </PROPERTYLIST>
  <PROPERTY NAME="server">Apache/2.0.40 (Unix) DAV/2</PROPERTY>
  <PROPERTY NAME="queryID">2</PROPERTY>
</PolicyValidatorQuery>
```

What properties a query contains

The table below outlines those query property tags that do not have any special-purpose plugins, but that can be evaluated using string operations. If you frequently use one of these query properties and would like a plugin for it, SelectAccess's extensible architecture allows you to build your own plugin. For details, see the *HP OpenView Select Access 6.0 Developer's Tutorial Guide*.

Table 39: Query properties syntax

Query property	**Description**	**Value**
`cert`	A property that sends information required by the certificate decision point.	A PEM-encoded X.509 digital certificate.
`client`	A property that sends information from the client software. **Note:** This data is not used by any of the standard server decision points.	The name and version number of the client software that initiated the request.
`dstHost` `srcHost`	A property that makes the host name explicit, when a single physical machine supports multiple virtual host names. **Note:** This data is not used by any of the standard server decision points.	The name of the host to which the request was sent.
`dstIP` `srcIP`	A property that retrieves destination IP information.	The IP address to which the request was sent.
`dstPort` `srcPort`	A property that retrieves destination port information. This data is required by the ports decision point.	The port to which the request was sent.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

Table 39: Query properties syntax (Continued)

Query property	Description	Value
`protocol`	A property that describes which format is used for transmitting data between the browser and server.	Any accepted protocol. For example, `http`, `https`.
`method`	A property that describes what HTTP header command was used to encapsulate the data.	Any valid HTTP header command. For example, `GET`, `POST`, `HEAD`.
`server`	A property that describes what kind of Web server is used.	Any supported Web/application server. For example, iPlanet Web server.

How the Policy Validator replies to the XML query

Replies are also formatted as XML. They can contain three things:

- An action telling the Enforcer plugin what to do next. Typical actions are:
  - Allow or Deny
  - Get more information by displaying a form
- Data that must be communicated to an application on the Web server. For example, a Java application might require information to personalize the experience for a user.
- Session information, namely a cookie or nonce that identifies the user on subsequent visits to one or more configured cookie domains. For details on cookies and nonces, see *Understanding nonces and cookies* on page 124.

For example, an XML response from the Policy Validator is shown in Code example 14. Notice how data is delimited by the following tag pairs:

- `<PolicyValidatorReply>` and `</PolicyValidatorReply>` delimit the entire response and all data it contains.
- `<PROPERTY NAME="`*`name`*`"></PROPERTY>` delimit the value for the property with `"`*`name`*`"`.

Code example 20: Example Policy Validator reply to the Enforcer plugin

```
<PolicyValidatorReply>
<PROPERTY NAME="queryID">2</PROPERTY>
  <PROPERTY NAME="authenticated_dn">cn=Lance
Mountain,ou=customer,dc=com,dc=user_isp</PROPERTY>
  <PROPERTYLIST NAME="personalization">
    <PROPERTY NAME="User">lmountain</PROPERTY>
    <PROPERTY NAME="Phone">504%2D2325</PROPERTY>
    <PROPERTY NAME="Fax">504%2D2399</PROPERTY>
    <PROPERTY NAME="DName">
cn%3Dlance%20mountain%2Cou%3Dlance%2Cdc%3Dcom%2Cdc%3Duser_isp%2Cdc</PROPERTY>
    <PROPERTY NAME="Groups">customer%20people</PROPERTY>
  </PROPERTYLIST>
  <PROPERTY NAME="login_time">Thu Feb 27 12:59:45 2003
</PROPERTY>
  <PROPERTYLIST NAME="authentication_server_types">
    <PROPERTY NAME="authentication_method">password</PROPERTY>
  </PROPERTYLIST>
  <PROPERTY NAME="action">ALLOW</PROPERTY>
</PolicyValidatorReply>
```

For more details on XML and how Select Access takes advantage of it, see *Understanding the importance of XML* in the *HP OpenView Select Access 6.0 Developer's Tutorial Guide*.

The Query utility

The Query utility is a command line application that sends queries to a Policy Validator. You can use the Query utility to do large test-runs against a Policy Validator. Use the Query utility to:

- Check a single query's outcome.
- Evaluate performance within your environment.
- Review simple and advanced authentication.

For details, see *Usage scenarios* on page 155. You can find the query program in the *`<install_path>/<bin>`* directory.

Command line syntax

On Windows, start the Query utility by entering one of the following:

- `query [options]` *`url`*
- `query [options]` *`file1.xml file2.xml`* `...`

On Unix, start the Query utility by entering one of the following:

- `./query [options]` *`url`*
- `./query [options]` *`file.xml file2.xml`* `...`

where:

- *`url`* is the fully qualified network location of the resource that is parsed and used to create the XML query. Entering only part of a URL results in the query failing, as all parts are needed by the Policy Validator. For example:
  `http://www.perftest.com:80/index.html`
  The parts of this URL are as follows:
  — `http` is the protocol.
  — `www.perftest.com` is the host.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

   — `80` is the port.
   — `index.html` is the resource.

- *`file.xml`* is the file that contains one or more queries in XML format.
- `[options]` can be almost any combination of the items listed in Table 40.

Table 40: Query program options

Options	Usage
`-a` *`user:password`*	Adds the username and password information into each query. Use this parameter to check the authentication process of the Policy Validator.
`-C` *`filename`*	Defines the certificate used. *`filename`* is the name and location of the PEM format certificate.
`-c` *`filename`*	Specifies an Enforcer plugin's configuration file. *`filename`* is the name and location of the configuration file. If no *`filename`* is provided, the default is used.
`-d`	Enables internal debugging. You can increment the debug level by one for each parameter you use. For example, -dd increments the level of debugging to level two (which enables tracing).
`-e` *`number`*	Defines the kind of result you expect to receive from the Policy Validator. Options supported are: • `0` — ALLOW • `1` — DENY • `2` — ERROR
`-f` *`number`*	**For Unix only.** Forks the query into the corresponding *`number`* of parallel processes.
`-h`	Shows parameter descriptions and help.
`-l` *`number`*	Specifies the number of times the queries are sent.

Table 40: Query program options (Continued)

Options	Usage
`-M`	Specifies that the query being sent is a management action (for example, that query contains information to allow the Policy Validator to re-configure itself).
`-m`	Enable nonce merging. If enabled, a nonce sent in a reply from the Policy Validator is merged into subsequent queries. This option is used to test the authentication process. This option can only be used with the -n option.
`-n`	Enable nonce support. To make reauthentication unnecessary, Select Access creates nonces to keep track of users who have already been authenticated. These nonces are generated by the Policy Validator's authentication plugin.
`-t` *`number`*	Specifies the number of threads inside each process. Each thread sends the request separately.
`-u` *`filename`*	A file that adds a series of usernames and passwords into each query. *`filename`* is the name and location of the user file. Use this parameter to check the authentication process of the Policy Validator.
`-V`	Returns the version number of the Query utility and exits.

Use `-f`, `-t`, and `-l` options for the Policy Validator stress testing.

You cannot combine some parameters with others. For example, you cannot use the `-a` parameter with the `-u` parameter, because it does not make sense to do so.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

Usage scenarios

Depending on your business environment and your goal for using this utility, the command line syntax varies. However, we have included some common scenarios below.

Testing a single query's outcome

To test a single query, run the Query utility at a debug level of two, and include the anticipated outcome of that query. For example, on Windows run the following command:

```
query -dd -e 0 http://www.mycompany.com:80/demo
```

This example creates a single query with one thread that is sent to the Policy Validator once.

Simple performance and stress testing

To stress test and measure Policy Validator performance, run the Query utility with forking (available on Unix only) and looping options for maximum results. For example, on Solaris run the following command:

```
./query -f 16 -l 2000 -e 0 http://www.mycompany.com:80/demo
```

This example creates 16 parallel processes with one thread each that are sent to loop the Policy Validator 2000 times. When the Query utility finishes running, it provides a summary as it exits. For example,

```
66206 queries per minute: 32000 queries in 29 seconds.
```

Simple authentication tests

To query a resource that has SelectID enabled, run the Query utility with the authentication option enabled. For example, on Windows run the following command:

```
query -dd -a jdoe:secretword http://www.mycompany.com:80/demo
```

This example creates a single query with one thread that gets sent to the Policy Validator for a single instance.

Advanced authentication tests with nonce support

To query a resource that has SelectID enabled and confirm that nonce support is functioning correctly, run the Query utility with the authentication and nonce support options enabled. For example, run the following command:

```
query -dd -n -m -a jdoe:secretword
http://www.mycompany.com:80/demo
```

This example creates a single query with one thread that gets sent to the Policy Validator. The results that are traced show that nonce support is enabled in the first query. The first Policy Validator reply then contained a new nonce. The Query utility subsequently adds that nonce into the second query. The second reply does not have a nonce because the Policy Validator knows the nonce is fresh.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

Appendix D

Policy Validator stability: setting file descriptor limits

Over the course of several releases, Policy Validator's stability has been improved by allowing the Policy Validator to handle a greater number of open connections (which includes Enforcer plugin connections). However, each platform has a limit on the number of simultaneous connections they allow. Table 41 summarizes these differences by platform.

Table 41: Simultaneous connection limitations by platform

Platform	Connection limit
HP-UX	60,000
All other platforms	8192

If you need to increase the number of simultaneous connections on your platform, you need to do so manually.

Increasing Unix connection limits for Policy Validator

In order to achieve a high number of connections you need root privileges to run the script that starts the Policy Validator on Unix platforms with the `ulimit` parameter. Currently, the scripts used to start the Policy Validator on these platforms set the limit of simultaneous connections on these platforms to 4096.

The initial nfile value is calculated by the system. Therefore, you may have a different value than the one we have specified.

Configuring the limit on Linux and Solaris

This is performed with the `ulimit -SHn 4096` command. For most deployments of Select Access, this limit is sufficient. However, if your deployment requires a higher or lower number, you can modify the number by editing this command in the startup script.

To change the limit in Policy Validator's startup script

1. Open the `<install_path>/validator` script.
2. Change the value of the `ulimit -SHn 4096` command.
3. Save the file and run this script to restart the Policy Validator with this new connection limit.

Configuring the limit on HP-UX

The default HP-UX kernel can only open about 920 files at once. This is a system-wide limit, shared between all processes on the system. Since the number of maximum connection can exceed the maximum limit on HP-UX, you need to set HP-UX's descriptor limit to a number that is much larger value than the maximum number of connections required for the Policy Validator.

To find the current number of open files vs. the current limit

You can find the current number of open files as well as HP-UX's current file descriptor limit, with the System Activity Reporter. Using the command `#sar -v 300` does the following:

- Monitors the system for five minutes.
- Report the results; look under the `file-sz` column for the values.

To increase the file descriptor limit

1. Run the System Administration Manager (sam).
2. Click the **Kernel Configuration** link.
3. In the window that appears, click the **Configurable Parameters** link.
4. Click on **nfile** and change the value of the kernel's `nfile` parameter to a more suitable number.

   For details, see the following documents on HP's Web site:

   – http://forums.itrc.hp.com/cm/QuestionAnswer/1,,0x4f6b402f24d5d61190050090279cd0f9,00.html

   – http://docs.hp.com/hpux/onlinedocs/939/KCParms/KCparam.Nfile.html

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

Index

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.

© Copyright 2000-2004 Hewlett-Packard Development Company, L.P.