HP OpenView Select Access

Integration Paper for BEA WebLogic Server v6.1

Software Version: 5.2

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

October 2003

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

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.

OpenView Select Access includes software developed by third parties. The software 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 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 offers.

You can 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

Contents

Chapter 1
About this Integration Paper

What is it about?

This Integration Paper describes how to integrate BEA WebLogic application server with HP OpenView Select Access.

An overview of this document's contents is listed in Table 1.

Table 1: Integration Paper overview

This chapter ...	Covers these topics...
Chapter 2, *Technologies overview*	• Introduces HP OpenView Select Access: what it is, what it does, and how it works. • Introduces BEA WebLogic application server: what it is and what integration issues exist.
Chapter 3, *Integration issues*	Describes what you need to do with BEA WebLogic application server and Select Access to integrate these technologies.

Who is it for?

This Integration Paper is intended to instruct individuals or teams responsible for the following:

- Integrating Select Access with their BEA WebLogic application server.
- Using Select Access to manage access to BEA WebLogic application server resources.

What does it assume you already know?

This Integration Paper assumes a working knowledge of:

- ***HP OpenView Select Access*—**Ensures that you understand how integration with BEA WebLogic application server affects the Select Access components.
- ***BEA WebLogic application server*—**Ensures that you understand how integration with Select Access affect the BEA WebLogic application server components.
- ***LDAP directory servers*—**Helps ensure that information in the Policy Builder is set up correctly.
- ***Web server and plugin technology*—**Combinations that are used to add a specific feature or service to a larger system. This helps you understand how different components of Select Access communicate with each other and with your existing network.

Related references

Before you begin to integrate Select Access with BEA WebLogic application server, you may want to begin by familiarizing yourself with the contents of the following documents:

- Hewlett-Packard, *HP OpenView Select Access v5.2 Installation Guide*, 2003 (`installation_guide.pdf`)
- Hewlett-Packard,*HP OpenView Select Access v5.2 Network Integration Guide*, 2003 (`network_integration_guide.pdf`)
- Hewlett-Packard, *HP OpenView Select Access v5.2 Policy Builder Guide*, 2003 (`policy_builder_guide.pdf`)
- Hewlett-Packard, *HP OpenView Select Access v5.2 Developer's Guide*, 2003 (`developer_guide.pdf`)
- Hewlett-Packard, Application/portal servers *Integration Papers*, 2003.

Chapter 2
Technologies overview

HP OpenView 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 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 the user's experience.

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 the appropriate 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

These different SSO methods means that users do not have to remember multiple passwords or PINs nor do they have to call your help desk because they have forgotten their passwords or PINs.

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 are predefined by the attributes you select. Typically, these attributes are the same 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 change to any number of output to:

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

Of all output choices, the Secure Audit server is considered to be 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, of which the resource listing is part of. 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.
- Policy 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 and your partners' sites.

These core components make for 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 with 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: 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/Resource 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.

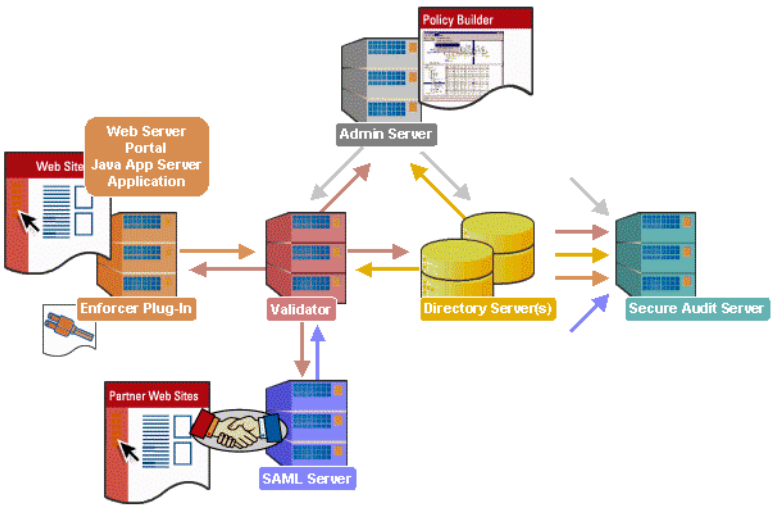

Figure 1: Select Access system architecture

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. Like the decision point plugin, this dialog box is known as a property editor, that allows security administrators to configure it.

- 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 display 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 Policy Validator returns to the Enforcer plugin's query.
- Additional Web/Application/Portal/Resource 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.

What is BEA WebLogic application server?

BEA WebLogic application server is an EJB application server. In a typical Select Access configuration with an EJB application server, all requests must be made through the Web server. When configuring your network, you do this by setting a user/resource policy against the Web server's entry on the resource tree. That way, when the user makes a request for the EJB application via the Web server, the Enforcer plugin queries the Policy Validator to determine whether or not the user is allowed to access the Web server and its resources. If the user is allowed, network requests proceed with the application server plugin which proxies a query to the application server. BEA WebLogic application server supports IIS, Apache, and Sun ONE Web servers. Depending on which Web server you use, you need to configure WebLogic differently.

Issues that affect BEA WebLogic application server

There are two main issues that affect BEA WebLogic application server:

- To enforce security across this dispersed architecture, you need to configure your servers so they always force resource requests through the Web server, rather than indirectly through the application server. This configuration ensures that requests are always handled by Select Access.
- The Enforcer plugin does not start automatically with the application server, therefore you need to create an IIS proxy `INI` file for your WebLogic application server.

The benefits of Select Access's solution

Integrating Select Access with BEA WebLogic application server offers the following main benefit:

- **Select Access EJB support follows the J2EE architecture**—a collection of Java technologies (EJB, JSP, Java API, and so on) that allows you to build a three-tier system, as shown in Figure 2. This system consists of a client, a server, and a database, which makes the deployment of EJB applications easier, and allows for enhanced transaction management.

The implementation of this server configuration is beyond the scope of this integration paper. This integration paper only describes how you can configure enforcer-protected Web servers with vendor-specific EJB application server plugins, so the plugins can proxy requests. It has been written under the assumption that your application server and Web server are installed on the same machine—and consequently the same platform. This combination makes the restriction of access to the application easier, while simultaneously facilitating communication.

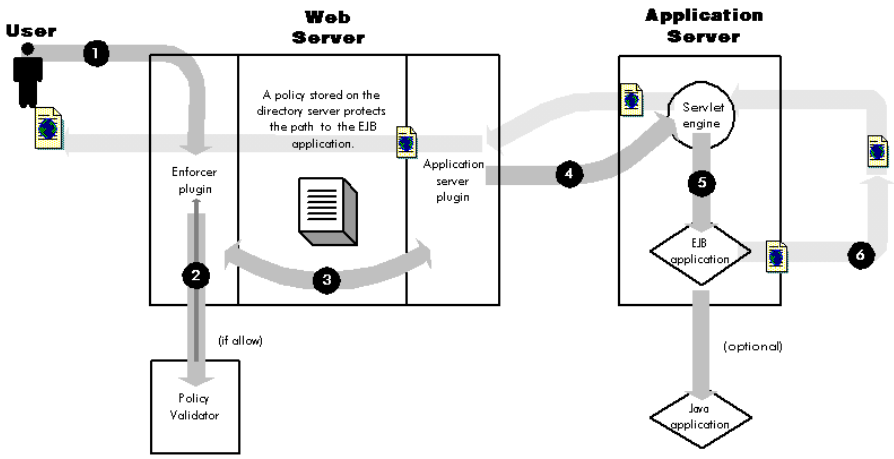

Figure 2: EJB support

Chapter 3
Integration issues

There are two sides to integrating Select Access with BEA WebLogic application server:

- **Configuring Select Access:** This requires that you:
  - Create a new network service for every URL that deploys the applications to which you want to control access.
  - Assign a deny access rule to restrict access to this URL.

For details, see *Configuring Select Access* on page 11.

- **Configuring BEA WebLogic application server:** This requires that you:
  - Settings to set up JSP support.
  - Create a new proxy-specific `INI` file to be used by WebLogic upon startup (IIS and Apache).
  - Copy the corresponding module (Sun ONE).
  - Restrict direct access further with a connection filter that restricts access based on IP address.

For details, see *Configuring BEA WebLogic application server* on page 12.

Configuring Select Access

Protecting your network does not just involve special configuration of the application server itself. Because the network is distributed, there are other facets to consider. Table 1 summarizes how you enable EJB application server support with this architecture.

Table 1: Setting up Select Access

Step ...	For details, see ...
1. Ensure that you have installed the enforcer plugin you need, depending on which server you have. **Note:** The Select Access IIS Setup Tool appends the IIS Enforcer plugin to the list of ISAPI filters. For WebLogic to work correctly, move the Select Access IIS Enforcer plugin filter before the WebLogic plugin filter, otherwise the Select Access-protected system has problems differentiating between application server resources and other resources on the network.	*Chapter 8, Configuring the Enforcer plugins* in the *HP OpenView Select Access v5.2 Installation Guide* AND/OR *Reordering plugins in your ISAPI filters list* on page 12
2. Create specific network resource entries in the Resources Tree for every URL that deploys the applications to which you want to control access.	*To create a new network service* in the *HP OpenView Select Access v5.2 Policy Builder Guide*
3. Assign a deny access rule to restrict access to this URL.	Chapter 8, *Controlling network access* in the *HP OpenView Select Access v5.2 Policy Builder Guide*

Reordering plugins in your ISAPI filters list

Launch the IIS Internet Services Manager and click the ISAPI filters tab. If WebLogic is listed before Select Access, use the up and down arrows to move the Select Access Enforcer plugin filter to the top.

Configuring BEA WebLogic application server

Because Select Access does not support WebLogic's built-in Web server, you need to use a Web proxy with a different Web server: IIS, Apache or Sun ONE. Depending on which Web server you use, you need to configure WebLogic differently, as suggested by Table 2.

To find out more about Web proxy configurations, please see the BEA WebLogic Server Administration Guide.

Step...	For details, see...
1. Configure BEA WebLogic application server to accept the Enforcer plugin as its proxy and bypass its built-in Web server. This restricts direct access to BEA WebLogic application server by letting the deny access rule prevent user access.	*Configuring Weblogic with an IIS Web server* on page 13 OR *Configuring WebLogic with a Sun ONE Web server* on page 15 OR *Configuring WebLogic with an Apache Web server* on page 16
2. Restrict direct access further with a connection filter that limits access to specific IP addresses only.	*Further restricting direct access to the WebLogic server* on page 17
3. Test the WebLogic application server with its plugins.	*Testing WebLogic with its proxies* on page 18

Configuring Weblogic with an IIS Web server

Configuring Weblogic with an IIS Web server involves setting up JSP support and then creating an INI file.

Most sites use a combination of JSP pages and paths for their Web applications. To proxy your JSP pages with the IIS plugin, you need to:

1. Map the extensions `.jsp` and `.wlforward`.
2. Define `iisforward.dll` as the filter.

To set up JSP support

1. Click **Start>Programs>Windows NT Option Pack>Microsoft Internet Information Server >Internet Service Manager**. The **Microsoft Management Console** window appears.
2. In the Console tree, right-click the Web site you are managing and select **Properties**. The **Default Web Site Properties** dialog box appears.
3. Select the **Home Directory** tab.
4. In the **Applications Settings** group, enable **Execute (including script)**, then click the **Configuration** button. The **Application Configuration** dialog box appears.
5. On the **App Mappings** tab, click the **Add** button . The **Add/Edit Application Extension Mapping** dialog box appears.
6. Click **Browse** to find the `iisproxy.dll` file.

   By default, this file is located in the following directory:

   `<install_path>\bin`

where *<install_path>* is WebLogic's default install path.

Once you've found `iisproxy.dll`, you need to do the following:

a. In the **Extension** field, enter `.jsp`.

b. Disable the **Script Engine** and **Check that file exists** check boxes.

c. Click **OK** to return you to the **App Mappings** tab of the **Application Configuration** dialog box.

7. On the **App Mappings** tab, click the **Add** button a second time to map a second extension. The **Add/Edit Application Extension Mapping** dialog box appears.
8. Click **Browse** to find the `iisproxy.dll` file again.

   Once you have found `iisproxy.dll`, you need to do the following:

   a. In the **Extension** field, enter `.wlforward`.

   b. Disable the **Script Engine** and **Check that file exists** check boxes.

   c. Click **OK**.
9. Click **OK** to close the **Application Configuration** dialog box. This returns you to the **Default Web Site Properties** dialog box.
10. Click the **ISAPI Filters** tab.
11. Click **Add**. The **Filter Properties** dialog box appears.
12. In the **Filter Name** field, type a name to describe the WebLogic filter.
13. Click **Browse** and select `iisforward.dll` as the WebLogic filter.

    By default, this file is located in the following directory:

    <install_path>\bin

    where *<install_path>* is the location where you installed the WebLogic application server.
14. Click **OK** to add the filter to the ISAPI **Filters** tab.
15. Click **OK** to close the **Default Web Site Properties** tab.

To create the INI file

1. In a text editor of your choice, create a new file and add the parameters defined in Table 2.

   By adding these parameters, you are creating a file similar to the example below which has path forwarding enabled:

```
# This file contains initialization name/value pairs
# for the IIS/WebLogic plug-in.

WebLogicHost=localhost
WebLogicPort=7001
ConnectTimeoutSecs=20
ConnectRetrySecs=5
WLForwardPath=/MyConverter,/MyHeaders
```

2. Save the file as `iisproxy.ini` and copy it to the same directory as `iisproxy.dll`.
3. Stop and restart the Web server to ensure it uses this new INI file.

Table 2: IIS proxy `INI` file parameters

This parameter ...	Takes this value ...	Description
`WebLogicHost`	`localhost`	**Required**. Enables HTTP requests forwarding to the WebLogic server host you define.
`WebLogicPort`	`7001`	**Required**. Defines the port at which the WebLogic server host parameter is listening for WebLogic connection requests.
`ConnectTimeoutSecs`	`20`	Sets maximum interval (in seconds) that the module tries to connect to the WebLogic server host. This parameter must be greater than the `ConnectRetrySecs` parameter.
`ConnectRetrySecs`	`5`	Sets the interval (in seconds) that the module sleeps between attempts to connect to the WebLogic server host.
`WLForwardPath`	`/MyConverter, /MyHeaders`	**Required** if you are proxying by path. Defines how requests are proxied. If `WLForwardPath` is blank, all requests are forwarded to `iis.proxy`. To forward any requests starting with a particular string, set `WLForwardPath`. The default value is "".

Configuring WebLogic with a Sun ONE Web server

Of all the proxies for WebLogic, the Sun ONE Web server proxy is the simpler one in terms of configuration. Configuring WebLogic proxy with an Sun ONE server requires that you:

- Copy the corresponding module.
- Add new lines and corresponding definitions to the `obj.conf` file.

To copy the module and edit the configuration file (Windows)

These actions are described in the procedure that follows.

1. Copy the WebLogic NSAPI plugin module, `proxy36.dll` (Windows) and `proxy36.so` (Unix), from `<install_path>\bin` into the directory where the NES is located.
2. Add the following lines to NES `obj.conf`:

For Windows:

```
Init fn="load-modules" funcs="wl_proxy,wl_init"\
shlib=PATH\proxy36.dll
Init fn="wl_init"
```

For Unix:

```
Init fn="load-modules" funcs="wl_proxy,wl_init"\
shlib=PATH\proxy36.so
Init fn="wl_init"
```

where:

- The first line loads the module and the functions used by the plugin. Replace *PATH* with the path to `proxy36.dll` (Windows) or `proxy36.so` (Unix).
- The second line tells the server to execute the `wl-init` function at server startup.

3. Define objects for each type of request that must be handled by the NSAPI plugin, by adding the following lines to the `<Object name=default>` section:

```
<Object name="weblogic" ppath="*/MyConverter/*">
Service fn=wl_proxy WebLogicHost=localhost \
WebLogicPort=7001
</Object>
<Object name="weblogic" ppath="*/MyHeaders/*">
Service fn=wl_proxy WebLogicHost=localhost \
WebLogicPort=7001
</Object>
```

4. Stop and restart the Web server to ensure it uses this new file.

Configuring WebLogic with an Apache Web server

If you are using the Apache Web server, the WebLogic server running on Solaris integrates with Select Access. To integrate Apache and WebLogic you need to edit Apache's configuration file.

To edit Apache's configuration file

1. Add the following lines to the `httpd.conf` file:

```
LoadModule weblogic_module PATH/mod_wl.so
AddModule mod_weblogic.c
```

where:

- The first line loads the module and the functions used by the plugin. Replace `PATH` with the path to `mod_wl.so`.
- The second line tells the server to install the module `mod_weblogic.c`.

2. Append the following lines:

   ```
   <Location /MyConverter>
     SetHandler weblogic-handler
   </Location>
   ```

   These lines indicate the plugin configuration options.
3. Comment out any lines that contain:

   ```
   ClearModuleList
   ```
4. Stop and restart the Web server to ensure it uses this new file.

Further restricting direct access to the WebLogic server

Even though you have already restricted access to the WebLogic application server via a deny access rule in the Policy Builder, WebLogic allows you to add connection filters that restrict access further, based on a set of IP addresses. WebLogic 6.0 includes some filters. By default, they are installed to the following location:

<install_path>\samples\examples\security\net

where *<install_path>* is WebLogic's default install path. For more information on WebLogic Connection Filters, see the section *Installing a Connection Filter* in the *WebLogic 6.0 Administration Guide*.

To install a connection filter that restricts access

1. Compile a file and move it to the following server classes directory. This allows WebLogic to locate the filter:

   <install_path>\config\<sitename>\serverclasses\

   where:

   — *<install_path>* is your default install path.

   — *<sitename>* is the name of your site you are managing.
2. Launch the **WebLogic Server Console** in your browser. In the main window, click the **Configuration** link under the **Security** heading. The **Security** screen appears.
3. Select the **General** tab and enter this interface name in the **Connection Filter** field:

   `weblogic.security.net.ConnectionFilter`.

   This interface is located in:

   <install_path>\samples\examples\security\net
4. Click **Apply** for your changes to be accepted.
5. Use the `SimpleConnectionFilter` file to create a customized filter file to define the connection rules. This file is located in the `serverclasses` directory described in step 1.
6. This configuration allows access using all protocols from the localhost and the machine IP address, but denies all other requests.

Append the following lines to the `SimpleConnectionFilter` file:

```
# allow localhost
127.0.0.1 allow
# allow assigned IP
10.10.10.75 allow
# deny everything else
0.0.0.0/0.0.0.0   deny
```

7. Stop and restart the Web server to ensure it uses this new file.

Testing WebLogic with its proxies

To ensure your application is fully secured, always test your application server with their proxy plugins. Select Access includes a sample EJB application, which is installed expressly for this purpose. HP has installed the WebLogic sample application to the following default directory:

<install_path>`\source\EJB\weblogic.`

This test application is a stateless session EJB application that performs simple currency conversion calculations. Parameters are then posted to a servlet via a JSP page, which contact the EJB application server for the results.

To test your WebLogic application server with your Web server proxy

1. Install the sample application. The sample EJB application consists of the following configuration files:
   - `Converter.jar` contains the EJB classes.
   - `MyConverter.war` contains the JSP pages and servlet classes.
   - `MyHeaders.war` contains the servlet to display the HTTP headers.
   - `ConverterServlet.java` contains the source for the converter servlet.
   - `HeadersServlet.java` contains the source for the headers servlet.

WebLogic 6.0 includes its own deployment descriptors. You have to extract class files for use in WebLogic 5.1.

2. Upon successful deployment of the `.jar` and `.war` files, test the pages by accessing the following URLs:
   - `http://<servername>/MyConverter/converter.jsp`
   - `http://<servername>/MyHeaders/HeadersServlet`

where <*servername*> is the name of the Web server proxy.

3. Test the Enforcer plugin on your Web server with the WebLogic application server by trying to access these URLs. These pages must be protected and you must not have direct access to it. If you are accessing these pages, you may have a problem with your WebLogic and/or its proxy configuration.