

## **HOW TO CONFIGURE PASS-THRU PROXY FOR ORACLE APPLICATIONS**

### **Overview of Oracle JInitiator**

Oracle JInitiator enables users to run Oracle Forms applications using Netscape Navigator or Internet Explorer. It provides the ability to specify the use of a specific Java Virtual Machine (JVM) on the client, rather than using the browser's default JVM.

Oracle JInitiator runs as a plug-in for Netscape Navigator and as an ActiveX component for Internet Explorer. Oracle JInitiator does not replace or modify the default JVM provided by the browser. Rather, it provides an alternative JVM in the form of a plug-in. Oracle provides two Jar files (f90all.jar and f90all\_jinit.jar). f90all.jar is a standard Jar file, and f90all\_jinit.jar is a Jar file with extra compression that can only be used with Oracle JInitiator.

Oracle JInitiator delivers a certified, supportable, Java Runtime Environment (JRE) to client desktops, which can be launched transparently through a Web browser.

Oracle JInitiator is Oracle's version of JavaSoft's Java Plug-in. The JavaSoft Plug-in is a delivery mechanism for a JavaSoft JRE, which can be launched from within a browser. Likewise, Oracle JInitiator is providing a delivery mechanism for an Oracle certified JRE, which enables Oracle Forms applications to be run from within a browser in a stable and supported manner.

### **How does JInitiator Work?**

The first time the client browser encounters an HTML file that specifies the use of Oracle JInitiator, it is automatically downloaded to a client machine from the application server. It enables users to run Oracle Application Server Forms Services and Graphics applications directly within Netscape Navigator or Internet Explorer on the Windows 98, NT, 2000, and XP platforms.

The installation and updating of Oracle JInitiator is performed using the standard plug-in mechanism provided by the browser. Oracle JInitiator installation performs the required steps to run Oracle Forms applications as trusted applets in the Oracle JInitiator environment.

### **Using Oracle JInitiator with Microsoft Internet Explorer:**

Oracle JInitiator leverages the Microsoft Internet Explorer extension mechanism for downloading and caching ActiveX controls and COM components. Using the HTML <OBJECT> tag, Web application developers can specify that ActiveX controls or COM components should run as part of a Web page. Such components include Oracle JInitiator.

When Internet Explorer first encounters an HTML file that has been modified to specify the use of Oracle JInitiator, Internet Explorer will ask the user if it is okay to download an ActiveX control signed with a VeriSign digital signature by Oracle Corporation. If the user clicks "Yes," Internet Explorer will begin downloading Oracle JInitiator. Oracle JInitiator will then run and use its parameters in the <OBJECT> tag to render the applet. The next time Internet Explorer encounters a Web page modified to support Oracle JInitiator, it will

seamlessly load and run Oracle JInitiator from the local disk, without user intervention.

**Using Oracle JInitiator with Netscape Navigator:**

Oracle JInitiator leverages the Netscape Navigator plug-in architecture in order to run inside the browser in the same way other plug-ins, such as QuickTime movies or Shockwave animations operate. Using the Netscape HTML <EMBED> tag, Web application developers can specify that plug-ins run as part of a Web page. This is what makes it possible for Oracle JInitiator to run inside the Web browser with minimal user intervention.

**Oracle Application Server Forms Services:**

Forms Services is a comprehensive application framework optimized to deploy Forms applications in a multi-tiered environment.

Forms Services Architecture:

- Client Tier - Web Browser
- Middle Tier - Application Server
- Database Tier - Database Server

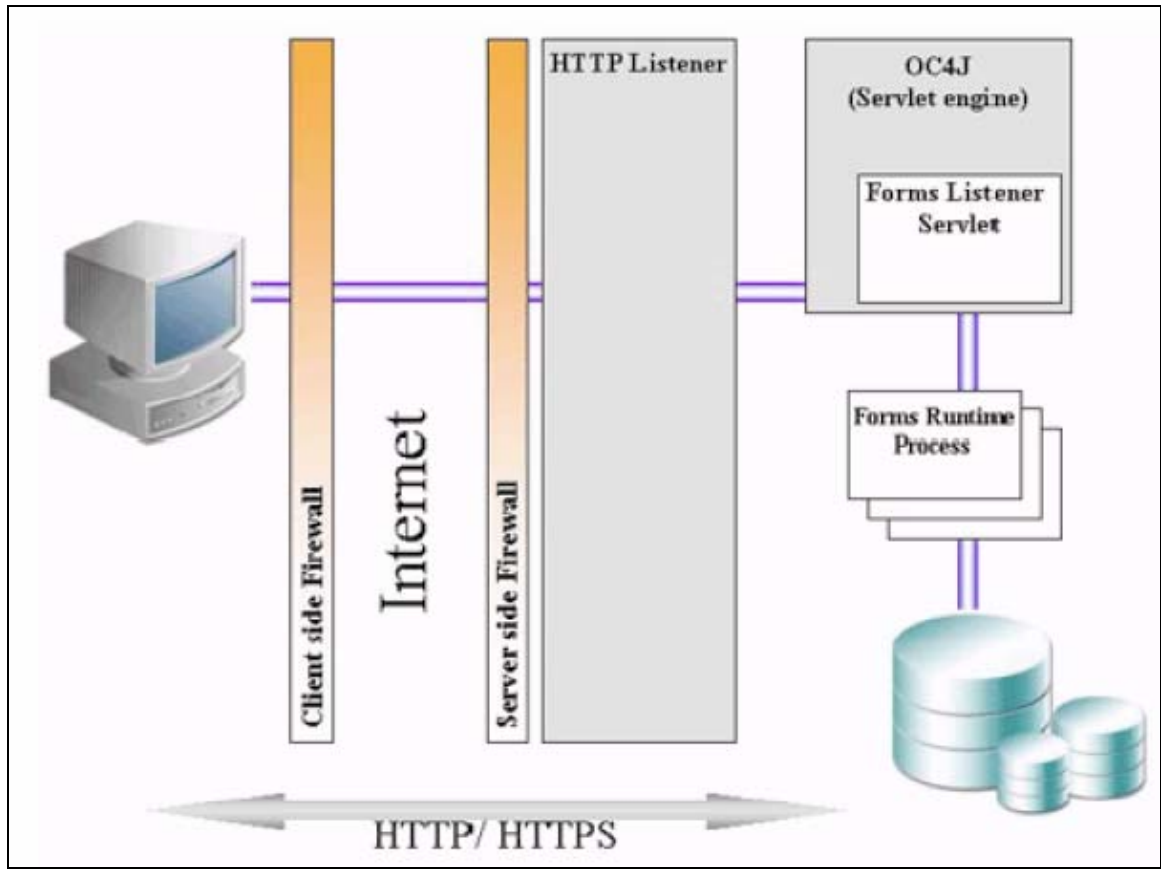
Forms Services Components:

- The Client - This resides on the client tier.
- The Forms Listener Servlet - Resides on the middle tier.
- The Forms Runtime Process - Resides on the middle tier.

**Forms Listener Servlet:** The Forms Listener Servlet acts as a broker between the Java client and the Forms runtime process. It takes connection requests from Java client processes and initiates a Forms runtime process on their behalf.

**Forms Runtime Process:** The Forms runtime process manages application logic and processing. It maintains a connection to the database on behalf of the Java client. It uses the same forms, menus, and library files that were used for running in client/server mode.

Architecture using Forms Listener Servlet



## Pass through Proxy (PTP)

### Overview

The pass-through proxy feature enables you to specify Web applications for which the IVE performs minimal intermediation. Unlike traditional reverse proxy functionality, which also rewrites only selective parts of a server response but requires network changes as well as complex configuration, this feature only requires that you specify application servers and the way in which the IVE receives client requests to those application servers:

- **Via an IVE port**

When specifying an application for the pass-through proxy to intermediate, you specify a port on which the IVE listens for client requests to the application server. When the IVE receives a client request for the application server, it forwards the request to the specified application server port. When you choose this option, you must open traffic to the specified IVE port on your corporate firewall.

- **Via external DNS resolution**

When specifying an application for the pass-through proxy to intermediate, you specify an alias for the application server hostname.

You need to add an entry for this alias in your external DNS that resolves to the IVE. When the IVE receives a client request for the alias, it forwards the request to the port you specify for

the application server. This option is useful if your company has restrictive policies about opening firewall ports to access the IVE. When using this option, we recommend that each hostname alias contains the same domain substring as your IVE hostname and that you upload a wild card server certificate to the IVE in the format: \*.domain.com. For example, if your IVE is iveserver.yourcompany.com, then a hostname alias should be in the format appserver.yourcompany.com and the wild card certificate format would be \*.yourcompany.com. If you do not use a wild card certificate, then a client's browser issues a certificate name check warning when a user browses to an application server, because the application server hostname alias does not match the certificate domain name. This behavior does not prevent a user from accessing the application server, however.

## Configuration

### Resource Policy -> Rewriting -> Pass-through Proxy tab

1. Use the **Rewriting > Pass-through Proxy** tab to write a Web resource policy that specifies Web applications for which the IVE performs minimal intermediation.
2. To create a pass-through proxy resource policy, you need to specify two things:
  - Which Web application to intermediate with the pass-through proxy
  - How the IVE listens for client requests to the application server
3. In the Web console, choose **Resource Policies > Web > Rewriting > Pass-through Proxy**.  
On the **Pass-through Proxy Policies** page, click **New Policy** and enter:
  - 1) A name to label this policy.
  - 2) A description of the policy (optional).
4. In the URL field, specify the application server hostname and the port used to access the application internally. Note that you cannot enter a URL in this field.
5. Choose the way in which you want to enable the pass-through proxy feature:
  - **Use virtual hostname**  
If you choose this option, specify a hostname alias for the application server. When the IVE receives a client request for the application server hostname alias, it forwards the request to the specified application server port in the **URL** field.  
**Important:** If you choose this option, you must also define the IVE name and hostname in the **Network Identity** section of the **System > Network > Internal Port** tab.
  - **Use IVE port**  
If you choose this option, specify a unique IVE port in the range 11000-11099. The IVE listens for client requests to the application server on the specified IVE port and forwards any requests to the application server port specified in the **URL** field.
6. In the **Action** section, specify the method for the IVE to use to intermediate traffic:
  - Rewrite XML
  - Rewrite external links
7. Click **Save Changes**.

8. On the **Pass-through Proxy Policies** page, order the policies according to how you want the IVE to evaluate them. Keep in mind that once the IVE matches the application requested by the user to an application specified in a policy's (or a detailed rule's) **Resource** list, it performs the specified action and stops processing policies.
9. If you select:
  - **Use virtual hostname**, you must also:
    - 1) Add an entry for each application server hostname alias in your external DNS that resolves to the IVE.
    - 2) Upload a wildcard server certificate to the IVE (recommended).
  - **Use IVE port**, open traffic to the IVE port you specified for the application server in your corporate firewall.

**Note:** If your application listens on multiple ports, configure each application port as a separate pass-through proxy entry with a separate IVE port. If you intend to access the server using different hostnames or IP addresses, configure each of those options separately; in this case, you may use the same IVE port.

### Examples

If your IVE is `iveserver.yourcompany.com` and you have an Oracle server at `oracle.companynetwork.net:8000`, you could specify these application parameters when specifying an IVE port:

```
Server: oracle.companynetwork.net
Port: 8000
IVE port: 11000
```

When the IVE receives Oracle client traffic sent to `iveserver.yourcompany.com:11000`, it forwards the traffic to `oracle.companynetwork.net:8000`.

Or, if you want to specify a hostname alias, you could configure the application with these parameters:

```
Server: oracle.companynetwork.net
Port: 8000
IVE alias: oracle.yourcompany.com
```

When the IVE receives Oracle client traffic sent to `oracle.yourcompany.com`, it forwards the traffic to `oracle.companynetwork.net:8000`.

When you choose to route client requests to the IVE based on a hostname alias, you must also add the IVE to your external DNS server. This option is useful if your company has restrictive policies about opening firewall ports to either internal servers or servers in the DMZ. Just as with the core intermediation engine, the pass-through proxy option offers increased security relative to the Secure Application Manager, because when enabled for an application, the IVE allows the client to send only layer-7 traffic directed to fixed application ports to the enterprise network. Use this option to enable the IVE to support applications with components that are incompatible with the content intermediation engine, such as Java applets in Oracle e-business suite applications or applets that run in an unsupported Java Virtual Machine.

**Note:** The pass-through proxy option works only for applications that listen on fixed ports and where the client does not make direct socket connections.

**Known Limitations on integration with IVE (PTP)**

1. We support the configuration ONLY if the Oracle Forms is configured in the "Forms Listener Servlet" mode. We do not support any other connection modes like HTTP, HTTPS or Socket. (Bug# 14000).

Reason: All other connection modes except "Forms Listener Servlet" try to talk to the Forms/Reports server directly. Also, the issue of trying to rewrite the content coming from the JInitiator Applet was fixed as per bug# 10500.

2. Starting from IVE version 4.1.1R3, we support Oracle Forms launched using the appsweb.cfg and formsweb.cfg. Prior to this we used to support only if it is configured using appsweb.cfg. (Bug# 23195)
3. Java applets in Oracle applications get the list of trusted root certificate authorities from JInitiator and not from the browser's certificate store. The IVE should have a valid and trusted certificate and it must be issued by one of these certificate authorities from Jinitiators.

**Data required for troubleshooting**

1. Screenshot of the error page.
2. Session recording log (Dsrecord.log) from IVE after clearing the browser cookies, cache and objects.
3. Tcpdump (using Ethereal) from the client workstation when the user is in the LAN and accessing the application directly (without IVE). Please ensure that browser cookies, cache and objects are cleared before capturing the dump.
4. Java Console log after clearing the Oracle JVM cache on items 2 (dsrecord through the IVE) and 3 (Tcpdump on client going direct to the application directly).

=====