Oracle Forms Services 10g Advanced Configurations

An Oracle White Paper
December 2006

Oracle Forms Services 10g Advanced Configurations

Oracle Forms Services 10g Advanced Configurations

INTRODUCTION

With the installation of Oracle Application Server 10g, Oracle Forms Services 10g and Oracle Reports Services 10g are automatically installed and configured for you to immediately deploy your Oracle Forms applications to the web. Oracle Forms Services has all its server and application specific settings configured in the *formsweb.cfg* and *default.env* configuration files, which are located in the forms90/server directory of the Oracle Application Server middle tier home. You use the formsweb.cfg file to create specialized application configurations to shortcut the Forms Services request URL. A typical Forms request URL for Oracle Forms 10g (9.0.4) looks like

This whitepaper covers advanced configurations for the Oracle Forms Services 10g (9.0.4) and Oracle Forms Services 10g R2 release (10.1.2). Note that Oracle Forms Services 10.1.2 no longer has the version digit in its configuration files and URL names.

http://server:port/forms90/f90servlet?config=appconf&parameter=value…[1]

Though there is nothing wrong in using the default configuration of Oracle Forms Services in a production environment, customizing the Forms configuration improves the Oracle Forms Services deployment platform and can provide additional security.

An example for using optimized custom Forms Services configuration is a reverse proxy that handles incoming Forms Services requests on an external HTTP Server located in the demilitarized zone, hiding the internal production server name and IP address from the outside world.

[1] Oracle Forms Services 10g (10.1.2) no longer has the version number within its virtual path and servlet name

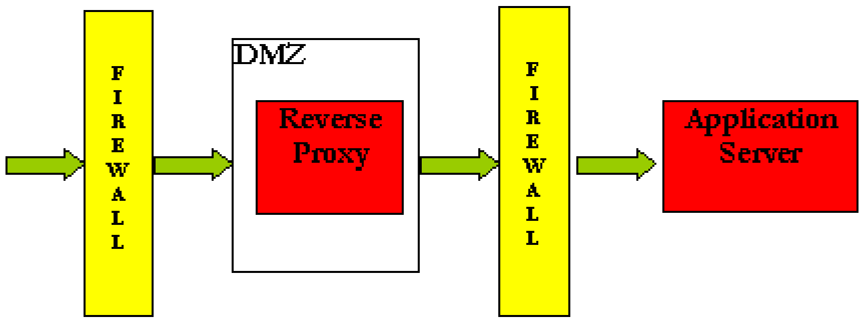


Figure 1: Proxy server architectural diagram

A second example where customizing the default Forms Services configuration becomes useful is when Oracle Forms Services applications need to be integrated with a J2EE architecture to enable Oracle Forms applications and J2EE applications to coexist and integrate in the same server architecture.

The following sections explain advanced Oracle Forms Services configuration settings.

CUSTOMIZING THE FORMS REQUEST URL

As mentioned earlier in this paper, the default Forms Services web request URL has the format of http(s)://server:port/forms90/f90servlet (or http(s)://server:port/forms/frmservlet in Forms 10g Release 2 (10.1.2))

Some enterprises use uniform URL patterns with their web deployed applications that, for example, include the name of the application. Such uniform URL patterns can be used for cosmetic reasons or for providing a more meaningful URL to application users.

One option to perform Oracle Forms URL customization is to use URL rewriting by leveraging the Oracle HTTP Server Redirect and Rewrite directives. In this configuration, your Forms application request URL is hidden from the application user. This approach is also recommended to integrate older Oracle Forms Services releases that are hosted on older versions of Oracle Application Server, with a newer version of Oracle Application Server acting as the front HTTP server.

Another option to customize your Forms Application request URLs is to change the Oracle Forms Services virtual path name and Oracle Forms Servlet name. You can change the default Forms Services request URL as follows:

1. Define an alias name for the Forms Servlet (f90servlet/frmservlet) as described in the Oracle Forms Services deployment guide, a part of the Oracle Application Server documentation library.

For example, changing the Forms Services Servlet name *f90servlet/frmservlet* to *myservlet* changes the Oracle Forms Services request URL to http://machine:port/forms90/myservlet

or

(http://machine:port/forms/myservlet)

Defining a Forms Servlet alias for the *f90servlet/frmservlet* is discussed in the section titled "Aliasing the Forms Servlet and Forms Listener Servlet".

2. Setup a custom OC4J container instance to deploy the Oracle Forms Services libraries to a different J2EE context root. This allows the Forms Services URL to be changed from

http://server:port /forms90/f90servlet

or (http://server:port/forms/frmservlet)

to

http://server:port/myvirtual_directory /f90servlet

or (http://server:port/myvirtual_directory/frmservlet).

This configuration is discussed in the section "Deploying Forms in a Custom OC4J Container".

Aliasing the Forms Servlet and Forms Listener Servlet

There are beneficial reasons for defining an alias name for the Forms Services Servlet:

- To customize the Forms Services request URL
- To use separate formsweb.cfg file for different applications
- To define different Servlet initialization parameters
- To comply with any security policies for your application, computer, or enterprise

Alias names for the Oracle Forms Services Servlet and the Oracle Forms Services Listener Servlet are defined in the web.xml deployment descriptor file of the Oracle Forms installation. The web.xml deployment file is a standard J2EE configuration file that you edit with any text editor. The default Oracle Forms Services installation deploys the Forms Servlet and the Forms Listener Servlet to the OC4J_BI_FORMS directory of the Oracle Application Server 10g middle tier installation. The web.xml deployment file is located in the J2EE\OC4J_BI_Forms\applications\forms90app\forms90web\WEB-INF

Or

(J2EE\OC4J_BI_Forms\applications\formsapp\formsweb\WEB-INF in Forms 10g Release 2 (10.1.2)) sub-directory of the Oracle Application Server middle tier installation.

The new Forms Servlet alias name is defined below the <web-app> element using the <servlet> and <servlet-mapping> elements. The following configuration defines a Forms Servlet alias myf90servlet for the Oracle Forms Servlet and myl90servlet for the Oracle Forms Listener Servlet.

```
<servlet>
        <servlet-name>myf90servlet</servlet-name>
        <servlet-class>oracle.forms.servlet.FormsServlet</servlet-class>
</servlet>
<servlet>
        <servlet-name>myl90servlet</servlet-name>
        <servlet-class>oracle.forms.servlet.ListenerServlet</servlet-class>
</servlet>

<servlet-mapping>
        <servlet-name>myf90servlet</servlet-name>
        <url-pattern>/myf90servlet*</url-pattern>
</servlet-mapping>
<servlet-mapping>
        <servlet-name>myl90servlet</servlet-name>
        <url-pattern>/myl90servlet*</url-pattern>
```

```
   </servlet-mapping>
```

Servlet initialization parameters, like the conFigFileName and default envFile, are added to the new aliases as follows:

```
    <servlet>
        <servlet-name>myf90servlet</servlet-name>
        <servlet-class>oracle.forms.servlet.FormsServlet</servlet-class>
            <init-param>
                <param-name>configFileName</param-name>
                <param-value><config file name></param-value>
             </init-param>
    </servlet>

  <servlet>
        <servlet-name>myl90servlet</servlet-name>
        <servlet-class>oracle.forms.servlet.ListenerServlet</servlet-class>
            <init-param>
                 <param-name>envfile</param-name>
                  <param-value><env file name></param-value>
            </init-param>
    </servlet>
```

To make Oracle Application Server aware of the new Forms servlet alias, you need to update the Oracle HTTP Server configuration file forms90.conf (or forms.conf in Forms 10g Release 2 (10.1.2)) to include the new servlet alias names. The forms90.conf (forms.conf) file is located in the forms90\server(forms\server) directory of the Oracle Application Server middle tier home.

```
        (forms90.conf)

        OC4JMount /forms90            OC4J_BI_Forms
```

```
OC4JMount /forms90/myf90servlet   OC4J_BI_Forms
OC4JMount /forms90/myf90servlet/* OC4J_BI_Forms
OC4JMount /forms90/myl90servlet   OC4J_BI_Forms
OC4JMount /forms90/myl90servlet/* OC4J_BI_Forms
```

(forms.conf)

```
OC4JMount /forms             OC4J_BI_Forms
OC4JMount /forms/myf90servlet   OC4J_BI_Forms
OC4JMount /forms/myf90servlet/* OC4J_BI_Forms
OC4JMount /forms/myl90servlet   OC4J_BI_Forms
OC4JMount /forms/myl90servlet/* OC4J_BI_Forms
```

You can use a text editor or the Oracle Application Server Control administration screen to edit the forms90.conf (forms.conf) file. For the changes to take effect, the Oracle HTTP Server component of the Oracle Application Server needs to be restarted.

Deploying Forms in a custom OC4J container

To build up two independent Oracle Forms Services instances, which may be required to clearly separate between a pre-production and a production stage on the same computer, you can create separate OC4J instances in Oracle Application Server.

Oracle Forms Services is setup in the new OC4J instance by deploying the forms90app.ear or (formsapp.ear in Forms 10g Release 2 (10.1.2)) file, which is located in the forms90\j2ee (forms\j2ee) directory of the Oracle Application Server home.

Using the Oracle Application Server Control web interface, create a custom OC4J container instance as follows:

Creating a customer OC4J instace

Start the Oracle Application Server Control for the Oracle Application Server middle tier by typing the following URL into the URL field of a browser.

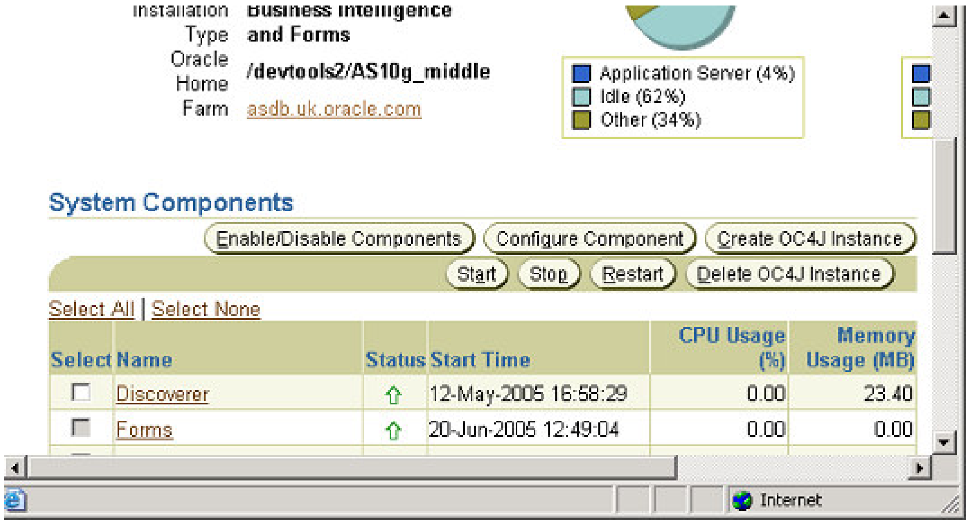
http://server.domain:1811

Note that the port number is installation specific and may differ from this example. You can look up the port number for the Application Server Control in the *portlist.ini* file, which is located in the /install directory of the Oracle Application Server middle tier home.

Authenticate with the Application Server control with ias_admin/<password>, where the <password> is the one provided during Oracle Application Server installation.

In the following page, select the Oracle Application Server middle tier installation.

1. To create a new OC4J instance, click the "Create OC4J instance" button on the Oracle Application Server Control website.

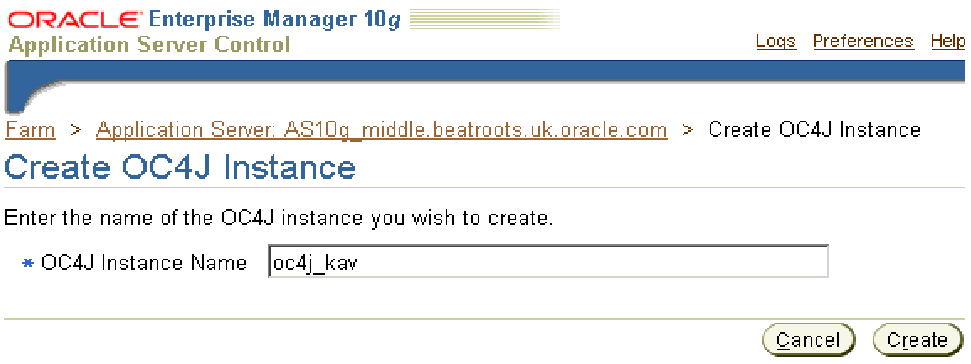


Installation Type: Business Intelligence and Forms
Oracle Home: /devtools2/AS10g_middle
Farm: asdb.uk.oracle.com

System Components

Select	Name	Status	Start Time	CPU Usage (%)	Memory Usage (MB)
☐	Discoverer	⇧	12-May-2005 16:58:29	0.00	23.40
☐	Forms	⇧	20-Jun-2005 12:49:04	0.00	0.00

2. Each OC4J instance must have a unique name, which is provided in the first dialog form. The name can be freely chosen. Click the "Create" button.



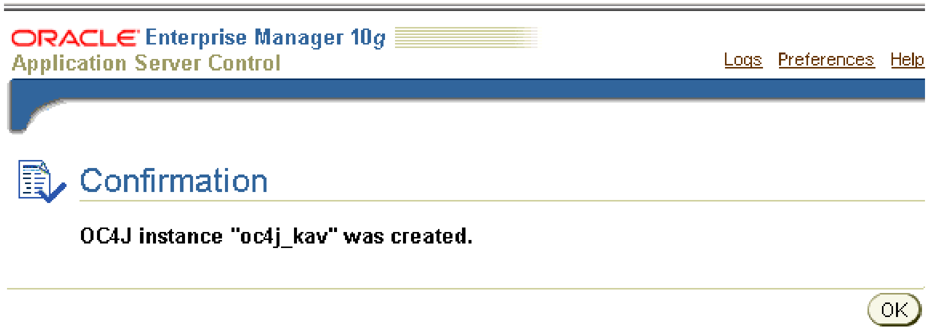
Farm > Application Server: AS10g_middle.beatroots.uk.oracle.com > Create OC4J Instance

Create OC4J Instance

Enter the name of the OC4J instance you wish to create.

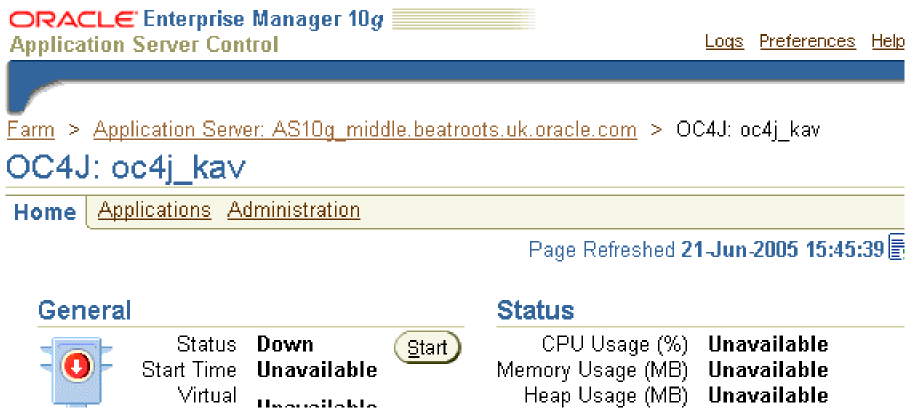
* OC4J Instance Name: oc4j_kav

On successful creation, a confirmation screen appears with the message "OC4J instance <name> was created". Click the "OK" button.



OC4J instance "oc4j_kav" was created.

3. The new OC4J instance appears in the list of available OC4J instance in the Oracle Application Server Control home page. Clicking on the OC4J instance name brings up the instance specific configuration page. The Oracle Forms Services EAR file is deployed following the "Applications" link next to the "Home" label.


ORACLE Enterprise Manager 10g
Application Server Control
Logs
Preferences
Help
Farm > Application Server: AS10g_middle.beatroots.uk.oracle.com > OC4J: oc4j_kav
OC4J: oc4j_kav
Home
Applications
Administration
Start

Page Refreshed 21-Jun-2005 15:45:39

General

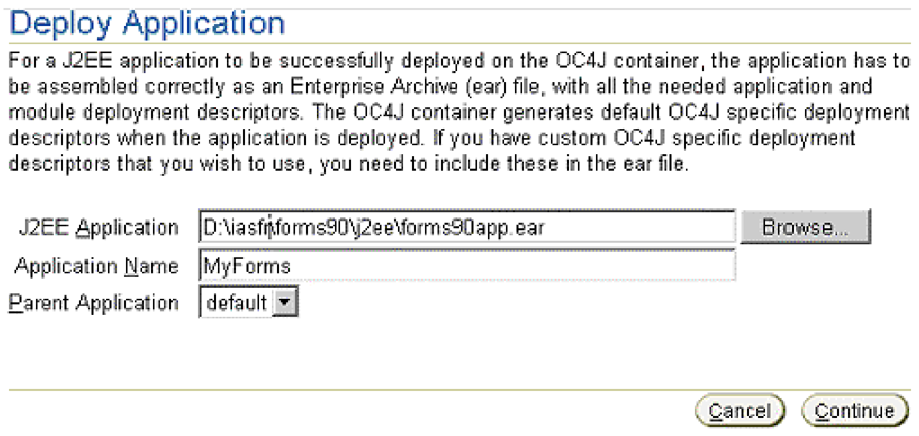
Status	Down
Start Time	Unavailable
Virtual	Unavailable

Status

CPU Usage (%)	Unavailable
Memory Usage (MB)	Unavailable
Heap Usage (MB)	Unavailable

4. Pressing the "Deploy EAR file" button starts the forms90app.ear(formsapp.ear) file deployment.

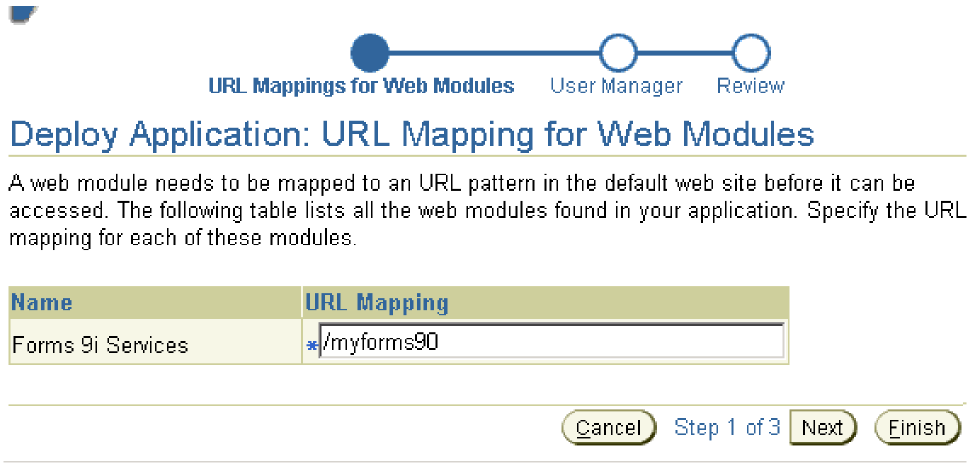
5. The next screen asks for three kinds of information: The location of forms90app.ear (formsapp.ear) file, the name of the application, the parent application, which should be set to "default". The name of the application can be anything.


Deploy Application

For a J2EE application to be successfully deployed on the OC4J container, the application has to be assembled correctly as an Enterprise Archive (ear) file, with all the needed application and module deployment descriptors. The OC4J container generates default OC4J specific deployment descriptors when the application is deployed. If you have custom OC4J specific deployment descriptors that you wish to use, you need to include these in the ear file.

J2EE Application	D:\iasfr\forms90\j2ee\forms90app.ear
Application Name	MyForms
Parent Application	default

6. The URL mapping that is defined on the following page defines the virtual access path to the Forms instance. The specified name is shown in the Oracle Forms Services request URL, similar to how */forms90* (/forms) shows in the default installation.

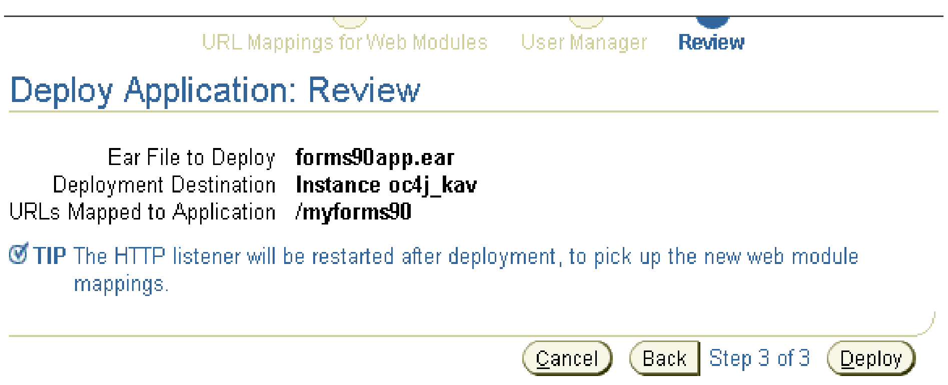


Deploy Application: URL Mapping for Web Modules

A web module needs to be mapped to an URL pattern in the default web site before it can be accessed. The following table lists all the web modules found in your application. Specify the URL mapping for each of these modules.

Name	URL Mapping
Forms 9i Services	* /myforms90

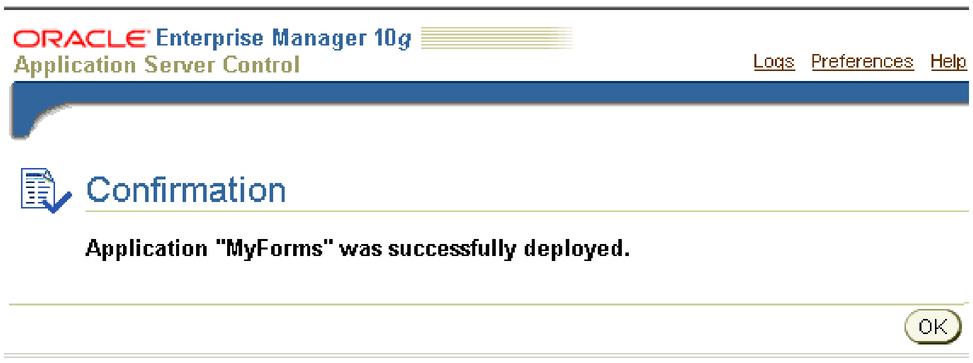
7. The following screens let you review your selections. Pressing the "Deploy" button on the last dialog page starts the Oracle Forms Services deployment process.



Deploy Application: Review

Ear File to Deploy	forms90app.ear
Deployment Destination	Instance oc4j_kav
URLs Mapped to Application	/myforms90

TIP The HTTP listener will be restarted after deployment, to pick up the new web module mappings.

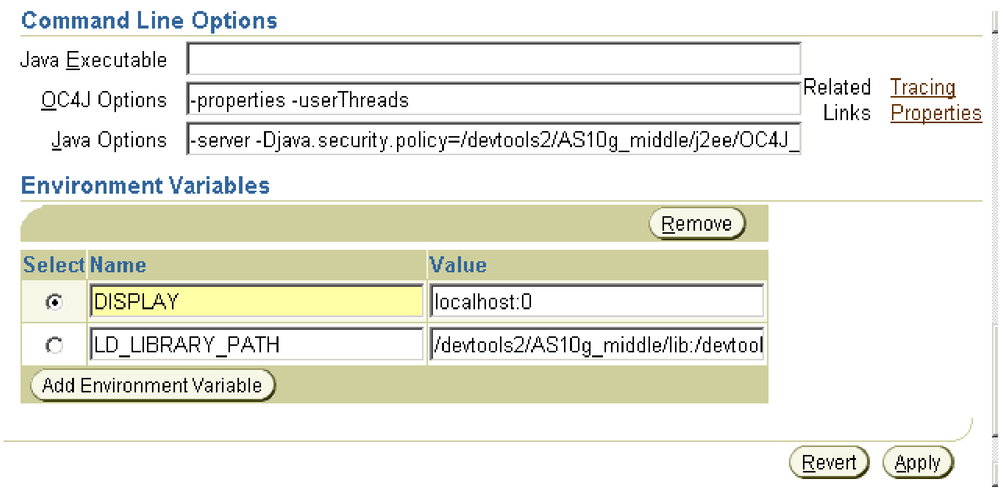


Confirmation

Application "MyForms" was successfully deployed.

8. The new OC4J instance needs to be configured with the same property entries as the OC4J_BI_Forms instance. You can obtain the instance specific properties by following the OC4J_BI_Forms link on the Oracle Application Server Control main page. The server properties are accessed by the "Server Properties" link on the "Administration" page.
   On the Microsoft Windows platform, the default OC4J_BI_Forms includes PATH and DISPLAY environment variables, while on Unix, the OC4J_BI_Forms instance properties include the DISPLAY, LD_LIBRARY_PATH or SHLIB_PATH or LIBPATH variables depending on the flavor of Unix used.

9. The properties of the newly created Oracle Forms Services instance are configured similar to the properties of the *OC4J_BI_FORMS* instance, following the "Administration" and "Server Properties" links. Press the "Add Environment Variable" button to create new properties for this specific instance. The "Apply" button writes the changed properties values to the configuration file.



Command Line Options

Java Executable	
OC4J Options	-properties -userThreads
Java Options	-server -Djava.security.policy=/devtools2/AS10g_middle/j2ee/OC4J_

Related Links: Tracing Properties

Environment Variables

Select	Name	Value
●	DISPLAY	localhost:0
○	LD_LIBRARY_PATH	/devtools2/AS10g_middle/lib:/devtool

10. Rename oc4j.properties located in ORACLE_HOME/j2ee/<new oc4j instance>/config directory. Copy oc4j.properties file from ORACLE_HOME/j2ee/OC4J_BI_Forms/config directory to ORACLE_HOME/j2ee/<oc4j instance>/config directory.

11. To allow the new OC4J instance to use Oracle Application Server Single Sign-On, add a new entry for the new oc4j instance`in the ORACLE_HOME/config/jazn-data.xml file. For example, say oc4j_kav is the new OC4J instance and MyForms is the name of the application specified (in step 8) during deployment of forms90app.ear (formsapp.ear), jazn-data.xml will have the following entry:

```
    <grant>
        <grantee>
            <codesource>
<url> file:ORACLE_HOME/j2ee/oc4j_kav/applications
      /MyForms/forms90web/WEB-INF/lib/f90srv.jar
                  </url>
            </codesource>
        </grantee>
        <permissions>
            <permission>
            <class>
```

```
                    oracle.ias.repository.schemaimpl.
                     CheckRepositoryPermission
              </class>
             <name>connectAs</name>
           </permission>
          </permissions>
        </grant>
```

For Forms 10g Release 2 (10.1.2):

```
        <grant>
           <grantee>
             <codesource>
    <url> file:ORACLE_HOME/j2ee/oc4j_kav/applications
        /MyForms/formsweb/WEB-INF/lib/frmsrv.jar
                 </url>
              </codesource>
           </grantee>
           <permissions>
              <permission>
              <class>
                    oracle.ias.repository.schemaimpl.
                     CheckRepositoryPermission
              </class>
              <name>connectAs</name>
               </permission>
            </permissions>
          </grant>
```

12. In formsweb.cfg file, modify serverURL as follows:
serverURL=/myforms90/l90servlet or (/myforms90/lservlet in Forms 10g Release 2 (10.1.2))

13. Start the new OC4J instance and access your Forms Application using URL
http:\\<machine>:port\myforms90\f90servlet?form=test.fmx

or

(http:\\<machine>:port\myforms90\frmservlet?form=test.fmx)

To test Oracle Application Server Single Sign-On functionality in the new OC4J instance, create a new user in Oracle Internet Directory (OID) and provide a Resource Access Descriptor for the user (RAD). Note down the resource name, like for example MYSSO, provided for the resource access descriptor.

In the formsweb.cfg, create a new configuration section with the same name as the resource name as follows:

```
[MYSSO]
 pageTitle=SSO Test Page
 form=ssotest.fmx
 ssoMode=true
```

Run the test form using this new configuration using the URL

http:\\<machine>:port\myforms90\f90servlet?config=MYSSO

or

(http:\\<machine>:port\myforms90\frmservlet?config=MYSSO)

This should immediately redirect you to the Oracle Application Server Single Sign-on dialogue Page with Username and Password fields. After providing the single sign-on username and password, the ssotest.fmx form will come up.
For more information on configuring Single Sign-on for Oracle

Forms, refer to the whitepaper "Oracle Application Server 10g Forms Single Sign-on", downloadable from OTN.

USING ORACLE FORMS RESTRICTED URL PARAMETERS

The Oracle Forms parameter *restrictedURLparams* is defined in the default configuration section and allows developers to specify parameters that a user should not be allowed to override in Oracle Forms Services request URL.

Oracle Forms parameters that are listed as a value of the *restrictedURLParameter* parameter in the formsweb.cfg file will be blocked if they appear in the Oracle Forms Services request URL. Any occurrence of a restricted URL parameter will lead to an error message for the user:

Restricted parameters <parameter name> cannot be specified in the URL.

The *restrictedURLparams* parameter can be defined in the default configuration section of the formsweb.cfg file, or for individual applications, located in the application specific configuration section.

The default configuration of *restrictedURLparams* is overridden in the application specific configuration. After installing Oracle Application Server and Oracle Forms Services 10g, the *restrictedURLparams* parameter is set to:

HTMLbodyAttrs, HTMLbeforeForm, pageTitle, HTMLafterForm, log, allow_debug, allowNewConnections

As a best-practices recommendation for security, Oracle suggests to add all possible Forms URL parameters as a value to the *restrictedURLparams* parameter in the default section of the formsweb.cfg file. In doing so, no URL parameter can be added by the application user to the Oracle Forms request URL. The application specific definition is then used to enable allowed URL parameters by overriding the default definition.

To ease administration, a new parameter, common_denied_parameters[2], can be added to the default section of the formsweb.cfg file. This parameter contains all Forms request parameters that should never appear in a request URL.

For example, the default section of the formsweb.cfg file can contain:

common_denied_parameters = <list all parameters that should be denied>

[2] The name can be freely chosen and is defined as common_denied_parameters only for this example

The *restrictedURLparams* parameter in the default section of the formsweb.cfg file can now be configured using the new common_demied_parameters variable

restrictedURLparams = %common_denied_parameters%

When overriding the default restrictedURLparams value in the application section of the formsweb.cfg file use the following entry:

```
[myApplication]
  form =
  restrictedURLparams = %common_denied_parameters%,<other restricted parameters
                                                                  for this application>
```

ACCESSING FORMS VIA A REVERSE PROXY

An Oracle Forms application may be secured using an Oracle HTTP Server reverse proxy. Oracle Forms Services is deployed on a server with no direct access to the end user and your Forms application can be accessed via the reverse proxy by:

1) Editing httpd.conf file to include the ProxyPass and ProxyPassReverse directives. Assuming that appserver is the name of the application server hosting the forms applciation, the following has to be included in httpd.conf:

```
ProxyPass /forms90/f90servlet
            http://appserver:port/forms90/f90servlet
ProxyPassReverse /forms90/f90servlet
            http://appserver:port/forms90/f90servlet
ProxyPass /forms90/java http://appserver:port/forms90/java
ProxyPassReverse /forms90/java  http://appserver:port/forms90/java
ProxyPass /forms90/l90servlet
             http://appserver:port/forms90/l90servlet
```

```
ProxyPassReverse /forms90/l90servlet
                 http://appserver:port/forms90/l90servlet
```

For Forms 10g Release 2 (10.1.2)

```
ProxyPass /forms/frmservlet
                 http://appserver:port/forms/frmservlet
ProxyPassReverse /forms/frmservlet
                 http://appserver:port/forms/frmservlet
ProxyPass /forms/java  http://appserver:port/forms/java
ProxyPassReverse /forms/java  http://appserver:port/forms/java
ProxyPass /forms/lservlet
                 http://appserver:port/forms/lservlet
ProxyPassReverse /forms/lservlet
                 http://appserver:port/forms/lservlet
```

2) Modifying the serverURL in the formsweb.cfg file as follows:

```
serverURL = /forms90/l90servlet/ (/forms/lservlet/)
```

(Note the extra trailing slash)

Note: Oracle HTTP Server reverse proxy should be stopped before making any changes to the configuration files

RUNNING FORMS WITH THE HTTP SERVER ON A SEPARATE SERVER

It is a common practice to separate the web listeners from other application server components for security reasons. Oracle HTTP Server and OC4J should reside on two different computers. Therefore, Oracle HTTP Server is remote to the OC4J_BI_Forms instance. Such a remote HTTP Server may be a standalone HTTP server or may be a part of an application server farm. For the purpose of this paper, let us consider a standalone HTTP server on host, say *standaloneserver*, and an application server instance on a different host, say *appserver*. Your Forms Application can be accessed via the remote HTTP server by:

1) Configuring OPMN of standalone HTTP Server

a) In the standalone HTTP Server home check opmn/conf/opmn.xml for the OPMN "remote" port.

For example, if opmn.xml looks like

```
<notification-server>
    <port local="6102"
            remote="6202"
            request="6005"/>
 [...]
```

the OPMN remote port is 6202

For more information, refer to the "Oracle HTTP Server Standalone Administrator's Guide Based On Apache 2.0 10g Release 2 (10.1.2)", whichcan be downloaded from OTN.

b) Copy ons.conf file from the application server instance (opmn/conf/ons.conf) to the opmn/conf directory of the standalone HTTP server instance.

The ons.conf file has the following format

```
nodes=<host_name | host_ip>[:port] [,<host_name | host_ip>[:port]] [, ...]
```

Edit the ons.conf file to include the IP address and opmn remote port of the standalone HTTP server. For example, if 6202 is the remote port, ons.conf will look like:

```
nodes=appserver:port,standaloneserver:6202
```

2) Configuring mod_oc4j of standalone HTTP Server

Note: This step only applies to a remote Standalone HTTP Server. This step is not required if the remote HTTP Server is a member of the same Application Server Farm as Oracle Forms Services because the opmn will be configured automatically. The opmn of a remote HTTP Server, which is a member of a different Application Server Farm to Oracle Forms Services, cannot be configured in this way.

Edit mod_oc4j.conf in Standalone http server and include the mount points for the remote OC4J_BI_Forms instance. For example, if "inst" is the name of the application server instance on host appserver, mod_oc4j.conf would look like:

```
Oc4jMount /forms90 instance://inst.appserver:OC4J_BI_Forms
Oc4jMount /forms90/java instance://inst.appserver:OC4J_BI_Forms
Oc4jMount /forms90/java/* instance://inst.appserver:OC4J_BI_Forms
Oc4jMount /forms90/jinitiator
          instance://inst.appserver:OC4J_BI_Forms
Oc4jMount /forms90/jinitiator/*
          instance://inst.appserver:OC4J_BI_Forms
Oc4jMount /forms90/f90servlet
          instance://inst.appserver:OC4J_BI_Forms
Oc4jMount /forms90/f90servlet/*
          instance://inst.appserver:OC4J_BI_Forms
Oc4jMount /forms90/l90servlet
          instance://inst.appserver:OC4J_BI_Forms
Oc4jMount /forms90/l90servlet/*
          instance://inst.appserver:OC4J_BI_Forms
```

For Forms 10g Release 2 (10.1.2)

```
Oc4jMount /forms instance://inst.appserver:OC4J_BI_Forms
Oc4jMount /forms/java instance://inst.appserver:OC4J_BI_Forms
Oc4jMount /forms/java/* instance://inst.appserver:OC4J_BI_Forms
Oc4jMount /forms/jinitiator
          instance://inst.appserver:OC4J_BI_Forms
Oc4jMount /forms/jinitiator/*
```

```
                instance://inst.appserver:OC4J_BI_Forms
    Oc4jMount /forms/frmservlet
                instance://inst.appserver:OC4J_BI_Forms
    Oc4jMount /forms/frmservlet/*
                instance://inst.appserver:OC4J_BI_Forms
    Oc4jMount /forms/lservlet
                instance://inst.appserver:OC4J_BI_Forms
    Oc4jMount /forms/lservlet/*
                instance://inst.appserver:OC4J_BI_Forms
```

3) Configuring orion-web.xml file of OC4J_BI_Forms instance

Edit orion-web.xml located in

ORACLE_HOME/j2ee/OC4J_BI_Forms/application-deployments/forms90app/forms90web directory and add the following lines:

```
<virtual-directory virtual-path=”/java”
 real-path=”ORACLE_HOME/forms90/java”/>
<virtual-directory virtual-path=”/jinitiator”
 real-path=”ORACLE_HOME/jinit”/>
```

For Forms 10g Release 2 (10.1.2)

Edit orion-web.xml located in

ORACLE_HOME/j2ee/OC4J_BI_Forms/application-deployments/formsapp/formsweb directory and add the following lines:

```
<virtual-directory virtual-path="/java"
 real-path="ORACLE_HOME/forms/java"/>
<virtual-directory virtual-path="/jinitiator"
 real-path="ORACLE_HOME/jinit"/>
```

Note: The standalone Oracle HTTP Server and OC4J_BI_Forms should be stopped before making any changes to the configuration files.

LOAD BALANCING ORACLE FORMS SERVICES

Oracle Forms services can be configured in different ways to balance the load of incoming Forms Services Requests across several Oracle Forms instances. In this paper, we will discuss the following two common configurations used for load balancing:

1) Two or more Oracle Forms Instances are in the same Application Server Farm and Oracle HTTP Server is used to distribute the load to the OC4J_BI_Forms instances in the Farm. For example, let us consider a Farm comprising three machines, MachineA has an Infrastructure installation, MachineB has a Business Intelligence and Forms mid tier and MachineC has another Business Intelligence and Forms mid tier. The HTTP Server in MachineA can be configured to distribute the load to the OC4J_BI_Forms instances in MachineB and MachineC. The mod_oc4j.conf file of HTTP Server can be edited to include OC4J mount points for MachineB and MachineC. For example, if inst1 and inst2 are the names of the application server instances on MachineB and MachineC respectively, then mod_oc4j.conf would look like:

```
Oc4jMount /forms90
          instance://inst1.MachineA:OC4J_BI_Forms,
          inst2.MachineB:OC4J_BI_Forms
Oc4jMount /forms90/java
          instance://inst1.MachineA:OC4J_BI_Forms,
          inst2.MachineB:OC4J_BI_Forms
Oc4jMount /forms90/java/*   I
          instance://inst1.MachineA:OC4J_BI_Forms,
          inst2.MachineB:OC4J_BI_Forms
```

```
Oc4jMount /forms90/jinitiator
        instance://inst1.MachineA:OC4J_BI_Forms,
        inst2.MachineB:OC4J_BI_Forms
Oc4jMount /forms90/jinitiator/*   I
        instance://inst1.MachineA:OC4J_BI_Forms,
        inst2.MachineB:OC4J_BI_Forms
Oc4jMount /forms90/f90servlet
        instance://inst1.MachineA:OC4J_BI_Forms,
        inst2.MachineB:OC4J_BI_Forms
Oc4jMount /forms90/f90servlet/*
        instance://inst1.MachineA:OC4J_BI_Forms,
        inst2.MachineB:OC4J_BI_Forms
Oc4jMount /forms90/l90servlet
        instance://inst1.MachineA:OC4J_BI_Forms,
        inst2.MachineB:OC4J_BI_Forms
Oc4jMount /forms90/l90servlet/*
        instance://inst1.MachineA:OC4J_BI_Forms,
        inst2.MachineB:OC4J_BI_Forms
```

For Forms 10g Release 2 (10.1.2)

```
Oc4jMount /forms
        instance://inst1.MachineA:OC4J_BI_Forms,
        inst2.MachineB:OC4J_BI_Forms
Oc4jMount /forms/java
        instance://inst1.MachineA:OC4J_BI_Forms,
        inst2.MachineB:OC4J_BI_Forms
Oc4jMount /forms/java/*
        instance://inst1.MachineA:OC4J_BI_Forms,
        inst2.MachineB:OC4J_BI_Forms
Oc4jMount /forms/jintiator
        instance://inst1.MachineA:OC4J_BI_Forms,
        inst2.MachineB:OC4J_BI_Forms
Oc4jMount /forms/jinitiator/*
        instance://inst1.MachineA:OC4J_BI_Forms,
        inst2.MachineB:OC4J_BI_Forms
Oc4jMount /forms/frmservlet
        instance://inst1.MachineA:OC4J_BI_Forms,
        inst2.MachineB:OC4J_BI_Forms
Oc4jMount /forms/frmservlet/*
        instance://inst1.MachineA:OC4J_BI_Forms,
        inst2.MachineB:OC4J_BI_Forms
Oc4jMount /forms/lservlet
        instance://inst1.MachineA:OC4J_BI_Forms,
        inst2.MachineB:OC4J_BI_Forms
Oc4jMount /forms/lservlet/*
        instance://inst1.MachineA:OC4J_BI_Forms,
        inst2.MachineB:OC4J_BI_Forms
```

The next step is to edit the orion-web.xml file of the OC4J_BI_Forms instances in MachineB and MachineC.

Edit orion-web.xml located in ORACLE_HOME/j2ee/OC4J_BI_Forms/application-deployments/forms90app/forms90web directory and add the following line:

```
<virtual-directory virtual-path=”/java”
real-path=”ORACLE_HOME/forms90/java”/>
<virtual-directory virtual-path=”/jinitiator”
real-path=”ORACLE_HOME/jinit”/>
```

For Forms 10g Release 2 (10.1.2)

Edit orion-web.xml located in

ORACLE_HOME/j2ee/OC4J_BI_Forms/application-deployments/formsapp/formsweb directory and add the following line:

```
<virtual-directory virtual-path=”/java”
  real-path=”ORACLE_HOME/forms/java”/>
<virtual-directory virtual-path=”/jinitiator”
   real-path=”ORACLE_HOME/jinit”/>
```

2) Two or more Oracle Forms Instances are in separate Application Server Farms and Oracle Webcache is used to distribute the load to the Farms. Web Cache can be used as a light weight load balancer to distribute HTTP Requests across two or more Application Server instances. In Web Cache releases prior to 10.1.2, the load balancing mechanism was very basic. Web Cache 10.1.2 has been greatly enhanced and improved to provide more robust, stateful load balancing. In this white paper, we will discuss how Web Cache 10.1.2 can be used to distribute the load across two or more Oracle Forms 10.1.2 instances. For example, let us consider a scenario where Web Cache 10.1.2 is installed in MachineA, a Business Intelligence and Forms mid tier instance is installed in MachineB and another Business Intelligence and Forms mid tier instance in MachineC. Web Cache 10.1.2 can be configured to load balance Oracle Forms Services by:

a) Configuring Oracle Web Cache for Load Balancing

1. Apply Patch 4569559 on top of Web Cache 10.1.2

2. Create a backup copy of the internal.xml file. Edit the internal.xml file. Locate the CALYPSOINTENRALPARAMS element

```
<CALYPSOINTERNALPARAMS>
<HEURISTICS CATELMFACTOR="0.0"/>
 <CACHE/>
<SEARCHKEY/>
<INVALIDATION/>
 <MEMORYMANAGER/>
 <PPC/>
<MISCELLANEOUS/>
 <OEMPERFTOOL/>
 </CALYPSOINTERNALPARAMS>
```

3. Add the LOADBALANCE subelement as follows:

```
<CALYPSOINTERNALPARAMS>
 <HEURISTICS CATELMFACTOR="0.0"/>
  <CACHE/>
  <SEARCHKEY/>
  <INVALIDATION/>
  <MEMORYMANAGER/>
  <PPC/>
  <MISCELLANEOUS/>
  <OEMPERFTOOL/>
  <LOADBALANCE ON="YES"/>
  </CALYPSOINTERNALPARAMS>
```

4. Save internal.xml and restart Web Cache

b) Mapping Web Cache Load Balancer to the Business Intelligence and Forms Mid Tiers

1. Log into Enterprise Manager and click on the Web Cache Link.
2. Click on the "Administration" link
3. Click on the "Origin servers" link and then press the “create button”. In the “create origin server screen”, add the host name and port of one of the mid tier instances (say

MachineB) and press “OK”. Similarly, add the host name and port of the second mid tier instance (MachineC)

4. After the origin servers are created, go back to the Web Cache administration page
5. Click on the link "Sites"
6. By default Web Cache is configured with a default site which uses the host name and listening port of the machine (Machine A in our case) on which web cache is installed. You can use this this default site definition or you can create a new custom site definition that suits your requirements by pressing the button "create". Any new site definition which you create should be above the default web cache site definition.
7. Edit the site and map it to the corresponding origin servers by moving them from "Available Origin Servers" box to "Selected Origin Servers" box. The site-to-server mappings for the custom site should be above the default site-to-server mappings.
8. Apply changes by pressing “OK” and restart Web Cache

c) Configuring Session Binding

1. Log on to Enterprise Manager
2. Click on the "Web Cache" link
3. Click on the "Administration" link
4. Click on the "Sites" link
5. Click on the "Default Session Binding" link
6. Click on checkbox "Enable Session Binding"
7. Select "JSESSIONID" from the “Session” drop down list
8. Select “OC4J Based” from the “Session Binding
9. Mechansim” drop down list and apply changes by pressing “OK”.

Note: This paper does not cover load balancing Forms 9.0.4 with Web Cache 9.0.4 due to the limited load balancing functionality in this release of Web Cache.

RUNNING ORACLE FORMS SERVICES OVER HTTPS

Communications containing sensitive information need protection. To protect data in transit, the commonly used technology is the Secure Socket Layer protocol (SSL), a technology that encrypts communication between the sending and the receiving endpoint. SSL can not only be used to encrypt messages, but also to verify that the message wasn't changed during transit. HTTP applications that use SSL are suffixed with an "s", HTTPS, and requested on a specific port, which by default is 443.

Oracle Forms Services applications are deployed to the web using the Oracle Application Server. Like any other web application that uses the Oracle HTTP server, Oracle Forms can leverage SSL transport layer security to protect its message communication.

For detailed information about configuring Oracle Forms Services for SSL, please read the whitepaper "Oracle Forms Services 10g: Configuring Transport Layer Security with SSL", which can be downloaded from OTN.

SUMMARY

This white paper details how the Oracle Application Server 10g platform can be effectively used to customize the deployment of Forms Applications, and have a tighter integration with the J2EE world.

Oracle Forms Services 10g Advanced Configurations
December 2006
Author: Kavitha Prakash
Contributing Authors:

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

Copyright © 2005, Oracle. All rights reserved.
This document is provided for information purposes only and the
contents hereof are subject to change without notice.
This document is not warranted to be error-free, nor subject to any
other warranties or conditions, whether expressed orally or implied
in law, including implied warranties and conditions of merchantability
or fitness for a particular purpose. We specifically disclaim any
liability with respect to this document and no contractual obligations
are formed either directly or indirectly by this document. This document
may not be reproduced or transmitted in any form or by any means,
electronic or mechanical, for any purpose, without our prior written permission.
Oracle, JD Edwards, and PeopleSoft are registered trademarks of
Oracle Corporation and/or its affiliates. Other names may be trademarks
of their respective owners.