Reconfiguring Oracle Forms Developer Web Cartridge Applications to CGI

An Oracle Technical White Paper
January 2000

INTRODUCTION

Traditionally, load balancing services in Oracle Forms Server were supplied via an Oracle Application Server (OAS) cartridge. If you wanted to deploy forms on the Web via a Common Gateway Interface (CGI) implementation in lieu of an OAS cartridge, load balancing was not an option.

With the release of Oracle Forms Server 6*i*, you can now use load balancing with forms applications that are deployed on the Web via a CGI implementation.

> *Note: With load balancing, when you approach the usage limits of your hardware, rather than upgrading or replacing a machine, you can simply add more machines to run your application and balance the load of server traffic across several machines.*

If you have already deployed Web-based Forms Developer applications via an OAS cartridge and you wish to switch to CGI, you will need to install Oracle Forms Server 6*i* and configure it for CGI. If you have already installed Oracle Forms Server 6*i*, you'll need to reconfigure it for CGI.

The purpose of this paper is to help those with existing cartridge-based implementations install or reconfigure Oracle Forms Server 6*i* from OAS cartridges to non-OAS CGI.

WHO SHOULD READ THIS PAPER?

This paper will be useful to you if the following statements apply to your deployment environment:

- You currently deploy Web-based Oracle Forms Developer applications.
- You own Oracle Forms Server 6*i*.
- You use OAS for Web server support.
- You deploy Web-based Oracle Forms Developer applications using OAS cartridges.
- You want to move from cartridge deployment to CGI.

ASSUMPTIONS

This paper describes the procedures you would use on a system running on either Windows NT or UNIX.

In the sections that discuss OAS, this paper describes user interface elements that appear in OAS versions 4.0 and higher. If you are using a version of OAS earlier than 4.0, refer to your OAS documentation for information on how to accomplish the OAS-related tasks described here.

OVERVIEW

COMPARING CARTRIDGE AND CGI

Cartridge and CGI implementations both require that you set server operational parameters that define values for such things as port numbers and locations of relevant files. The difference is in where you set them. In OAS, you open the OAS Manager and navigate to various destinations to set parameters for different deployment entities. In Oracle Forms Server, configuration complexity is more centralized. You set many operational parameters automatically through configuration choices you make during installation. You can revise and set additional operational parameters in Oracle Forms Server's formsweb.cfg file, which is created during installation.

Note: The formsweb.cfg file is discussed in detail, starting on page 12.

Cartridge and CGI implementations both produce an HTML file on-the-fly that is rooted in a standard base HTML file. In a cartridge implementation, the HTML file is created through a combination of the cartridg.html file, cartridge configuration settings, and the application's URL. In a Forms Web CGI implementation, the HTML file is created through a combination of the base.htm or basejini.htm file, the formsweb.cfg file, and the application's URL.

In both cartridge and CGI base HTML files, you can define a parameter with a variable and then define the variable value in the application's cartridge settings (OAS), the formsweb.cfg file (Oracle Forms Server 6*i*), or via a query string in the application's URL (both OAS and Oracle Forms Server).

The major differences between cartridge and CGI implementations are in the types of services and level of performance offered through your non-OAS Web server (as compared to those offered through OAS), the broader range of operational parameters now available through Oracle Forms Server 6*i*, and the vastly simplified process of setting forms parameters via Oracle Forms Server installation and the formsweb.cfg file.

RECONFIGURATION STRATEGIES

Note: This section provides a high-level overview of the reconfiguration process. It is suitable for users who have a technical understanding of OAS, Oracle Forms Server, base HTML files, and the like. For a more detailed description of the entire process, refer to the remaining sections in this paper.

There are two basic strategies for reconfiguring cartridge deployments to CGI:

- Keep everything the same, replicating cartridge parameters in the formsweb.cfg file.
- Use the default Oracle Forms Server 6*i* installation.

The first strategy is appropriate for users with complex base HTML files, that is, files that contain much extraneous text, images, and other objects in addition to the Forms applet tags. The second strategy is appropriate for users with simple base HTML files.

Strategy for Users with Complex Base HTML files: Keep Everything the Same

1. Stop all instances of OAS you will no longer use.
2. Install Oracle Forms Server 6*i*.
3. Install the WebDB Listener or some other CGI-capable Web server. (The Oracle WebDB Listener is provided at no cost with Oracle Forms Server 6*i*. It is offered as an installation option when you install Oracle Forms Server 6*i*.)
4. In the formsweb.cfg file, reproduce the parameters that were used for cartridge configuration.

   To locate current OAS cartridge parameters, launch OAS and navigate to each forms application's Cartridge Configuration folder. Within each folder, click Cartridge Parameters. This displays the OAS cartridge parameter settings. Additionally, you will find parameter settings in the base HTML file(s) you created for Forms cartridge applications.

5. If you were using several OAS cartridge definitions for the Forms cartridge (that is, you were using several base HTML files), define separate configuration sections in the formsweb.cfg file—one for each cartridge. (With both strategies, the better practice is to create new configuration sections for cartridge parameters in the formsweb.cfg file rather than specify the parameters at the start of the formsweb.cfg file, outside a named section.)
6. For a non-OAS Web listener, define the same virtual paths you used with OAS. (The installer does this task for you when you choose to install the Oracle WebDB Listener.) Add a new virtual path for CGI scripts that points to the directory containing the Forms CGI (ifcgi60.exe on NT, f60cgi on UNIX), as follows:

   virtual_path_name = /dev60cgi/

NT: physical_path = %ORACLE_HOME%\tools\web60\cgi
UNIX: physical_path = $ORACLE_HOME/tools/web60/cgi

7. Change all the URL's you use to run forms to point to the CGI rather than the cartridge. For example, if the original URL was:

   http://servername.my.domain.com/developerforms/forms60cart?module=emp.fmx

   It should become:

   NT: http://servername.my.domain.com/dev60cgi/ifcgi60.exe?config=myconfig&
   module=emp.fmx
   UNIX: http://servername.my.domain.com/dev60cgi/f60cgi?config=myconfig&
   module=emp.fmx

   In this example, "myconfig" is the name of the configuration section you defined in the formsweb.cfg file that contains the parameters equivalent to your old cartridge parameters.

Strategy for Users with Simple Base HTML Files: Use the Default 6*i* Installation

If your cartridge implementation used simple base HTML files, your reconfiguration to CGI can easily benefit from the default configuration that is created automatically during Oracle Forms Server 6*i* installation.

1. Stop all OAS instances you will no longer use.
2. Install Oracle Forms Server 6*i*.
3. Install the WebDB listener or some other CGI-capable Web server. (The Oracle WebDB Listener is provided at no cost with Oracle Forms Server 6*i*. It is offered as an installation option when you install Oracle Forms Server 6*i*.)
4. Configure virtual paths with your Web listener for use with Oracle Forms Server. (The installer does this task for you when you choose to install the Oracle WebDB Listener.)
5. Adapt the URLs you used to run your forms to achieve the same effect as you had with the cartridge. Use the parameters that are defined for you in the formsweb.cfg file. These allow you to change just about every conceivable HTML and Forms Applet parameter value by specifying the value in the application's URL. The same URL will work for users of the AppletViewer, a Web browser in combination with Oracle JInitiator, or Internet Explorer 5.0.
6. Use the runform.htm file to experiment with different parameter settings in the application's URL. For example, this might be the URL you would use to run a form with a page title "My Form," a page width of 400, and a page height of 550:

   NT: http://servername.my.domain.com/dev60cgi/ifcgi60.exe?pagetitle=
   My+Form&width=400&height=550
   UNIX: http://servername.my.domain.com/dev60cgi/f60cgi?pagetitle=
   My+Form&width=400&height=550

In these examples, the question mark signals the start of the query string in the application URL. The query string specifies the values for the pagetitle, width, and height parameters.

In Windows environments, you can get to the runform.htm file via the "Run a Form on the Web" shortcut, which is created by the installation process. In UNIX environments, specify a path to runform.htm, for example:
http://servername.my.domain.com/dev60html/runform.htm.

RECONFIGURING FORMS WEB CARTRIDGE TO CGI

Take these steps to reconfigure forms deployments from OAS cartridge to non-OAS CGI:

1. Stop the OAS Web Listener instances you will no longer use.
2. Install Oracle Forms Server 6*i*.
3. Install the WebDB Listener or some other CGI-capable Web server. (The Oracle WebDB Listener is provided at no cost with Oracle Forms Server 6*i*. It is offered as an installation option when you install Oracle Forms Server 6*i*.)
4. Perform the additional configuration steps that are specified in the installer-generated dev6iconfig.txt file.
5. Configure the Oracle Forms Server formsweb.cfg file.
6. Optionally, configure the Oracle Forms Server base.htm and basejini.htm files.
7. Broadcast the application's URL.

The rest of this paper discusses these steps in detail.

STEP 1: STOPPING OAS WEB LISTENER INSTANCES

There are two scenarios for stopping OAS:

- Stop it completely.
- Stop only specific instances while OAS continues to support other instances.

This section provides information for both scenarios.

STOPPING OAS COMPLETELY

Use this technique if you wish to stop using all services offered through OAS:

1. Launch OAS.
2. Open the OAS Manager.
3. Navigate to the top-level site of the OAS installation.

4. Select **All**.
5. Click the **Stop** button.

STOPPING SPECIFIC INSTANCES OF OAS

Use this technique if you wish to stop only some OAS HTTP Listeners and leave others running:

1. Launch OAS.
2. Open the OAS Manager.
3. Navigate to HTTP Listeners.
4. Select those HTTP Listeners running on ports you are planning to convert from cartridge to CGI.

   Note: If you plan to install the Oracle WebDB Listener, you must stop any OAS HTTP Listener that is running on port 80.
5. Click the **Stop** button.

STEP 2: INSTALLING ORACLE FORMS SERVER 6*i*

The Oracle Forms Server 6*i* installer offers many automatic configuration options during installation. The options you choose will depend on your needs and your environment. If your deployment needs change over time, the installer is “re-entrant.” That is, you can rerun the installer and change earlier configuration decisions.

This section discusses a multiple-machine installation with load balancing. It is written with the assumption that you will accept default configurations for Oracle Registry or Environment Variables and for Forms Server Startup Parameters (NT only). Information about reconfiguring these components is not included within the scope of this paper.

Note: If you are using Windows NT, before you run the installer, make sure you have administrative privileges on the Windows NT workstation where you will install the server.

To install Oracle Forms Server in a multiple-machine configuration with load balancing:

1. Shut down any active applications.
2. Insert the Oracle Forms Server installer CD.
3. Run the Oracle installer.
4. In the Oracle Installation Settings dialog box, elect a language in which to run Oracle Forms Server.
5. In the Oracle Installation Settings dialog box, enter your company name, the name of and path for the Oracle Home directory, and the language in which to run Oracle Forms Server.

   For name and location, we strongly recommend that you accept the defaults.

6. The installer presents a series of options. Your selections inform the installer how to configure your server. These are discussed in the next sections.

   Note: You'll find more information about installing Oracle Forms Server in Getting Started, *included with the Oracle Forms Developer 6i online manuals.*

INSTALLING FOR WEB CGI IMPLEMENTATION AND LOAD BALANCING

This section provides an example of the choices you would make if you were installing Oracle Forms Server in a multiple-machine configuration and intended to implement Web CGI and load balancing.

> *Note: The options presented in this section are for an NT server environment. For UNIX, the options vary slightly, but this information should, nonetheless, see you through.*

Option and Recommendation	Description
What product do you wish to install. **Choose:** Oracle Forms Server	This will reduce the number of installation decisions you must make to those related to Oracle Forms Server and its supporting components.
What type of installation would you like to perform? **Choose:** Typical	This will start you down an automated pathway to installing Web deployment components.
What type of deployment installation do you want ? **Choose:** Forms Server for Web deployment	Installs and configures the necessary components to run Oracle Forms Developer applications as Web applications on one or more server machines.
What type of server will this be? **Choose:** Part of a multiple machine configuration	Installs Oracle Forms Server as part of a multiple-machine configuration. Use this option when you also plan to use load balancing.
What type of node will this machine be? **Choose:** Primary Node when you are installing Oracle Forms Server on the machine where you will also install a Web server. Choose Secondary Node for all other installations.	When there are multiple machines, one machine acts as the Primary Node. It must have a Web listener running on it. It will receive and process all URL requests to execute applications. It routes each request to the machine that will actually execute the application. This could be a Secondary Node or the Primary Node, if Forms runtimes are available on that machine.
Where should Oracle Forms applications run? **Choose:** On the secondary nodes and this primary node	Select this option to allow Oracle Forms Developer applications to run on this Primary Node. In this case, the Forms Server will be installed and configured on this node along with the other Primary Node components, including load balancer components and Forms Runtimes.
Web Listener **Choose:** Either WebDB or another Web listener that supports CGI,	You need a Web listener (HTTP daemon) to run Oracle Forms Developer applications on the Web. For your convenience, the Oracle WebDB Listener, a light-weight listener that supports CGI, is provided with Oracle Forms Developer. Alternatively, you may

whichever you prefer.	use any other Web listener that supports CGI (Common Gateway Interface). If you choose a Web listener other than WebDB, after you install Oracle Forms Server, you will need to define some virtual paths. See page 12 for more information. If you choose WebDB Listener, you will also be asked to specify Data Access Descriptor (DAD) configuration information. DAD is used to capture database connection information for an application.

COMPONENT INSTALLATION OPTIONS

Some Oracle Forms Server installation options branch off into additional choices. This section provides details on configuring these additional choices during installation.

Deployment Component and Options	Recommended Action
Forms Server parameters Enter parameter values for the Forms Server. **Options:** Forms Server Port: [9000] Protocol: Sockets (default) HTTP	**Recommendation:** Accept the defaults. You may accept or modify the startup parameter values for the Forms Server. **Forms Server Port:** Enter the TCP/IP port number on which the Forms Server will listen for form execution requests. The displayed value need only be changed if another program already uses it. **Protocol:** This specifies the method to be used to communicate between the Forms runtime engine and the Forms Java Applet (which runs in the user's Web browser). Change the value to HTTP only if the communication needs to pass through a firewall. For example, select HTTP if this machine is inside a firewall but your Forms applications need to be available to users outside the firewall.
Load Balancer Server Parameters Enter the port numbers to be used by the Forms load balancer server. **Options:** Data port: [9010] Request port: [9020]	**Recommendation:** Accept the defaults. You may accept or modify the startup parameter values for the load balancer server. These are TCP/IP port numbers. Change a default port number only if the default number is already being used by another program. **Data port:** Enter the TCP/IP port number on which to listen for load data from the load balancer client processes (which run on Secondary Nodes). **Request port:** Enter the TCP/IP port number on which to listen for requests for the "least loaded host" made by the Forms Web CGI.
Load Balancer Client Parameters Enter the full host name of the Primary Node, and the data port number for the load balancer server running on that node. **Options:**	**Recommendation:** Accept the defaults. You may accept or modify the startup parameter values for the load balancer client. These tell it how to connect to the load balancer server. **Data host:** Enter the full host name of the machine that is the Primary Node (the machine on which the load balancer Server is

Data host: [ ]	running) in this multiple-machine configuration.
Data port: [9010]	**Data port:** Enter the TCP/IP port number on which the load balancer server is listening for load data.

STEP 3: INSTALLING YOUR CGI-CAPABLE WEB SERVER

The Oracle Forms Server 6*i* installation package includes the Oracle WebDB Web Listener. If you choose WebDB for Web server support, the Oracle Forms Server installer will automatically configure it for you. If you choose a Web listener other than WebDB, you must perform some additional Web server configuration. Specifically, you must:

- Specify a hostname.
- Specify a port number. Most Web servers use port 80.
- Create virtual paths that map to the physical directories listed in the following table. The Oracle Forms Server installer creates these physical directories. They contain files that are used by Forms.

Virtual Path	Physical Directory	Description
/forms60java	**NT:** %ORACLE_HOME%\Forms60\java **UNIX:** $ORACLE_HOME/Forms60/java	Forms Java files
/dev60html	**NT:** %ORACLE_HOME%\Tools\web60\html **UNIX:** $ORACLE_HOME/Tools/web60/html	Starter HTML files for running forms
/dev60cgi	**NT:** %ORACLE_HOME%\Tools\web60\cgi **UNIX:** $ORACLE_HOME/Tools/web60/cgi	CGI executables
/jinitiator	**NT:** %ORACLE_HOME%\Jinit\ **UNIX:** $ORACLE_HOME/Jinit/doc	JInitiator (for download)
/dev60temp	**NT:** %ORACLE_HOME%\Tools\web60\temp **UNIX:** $ORACLE_HOME/Tools/web60/temp	Forms temporary files

STEP 4: COMPLETING ADDITIONAL CONFIGURATION

Some additional configuration steps may be required once you finish installing Oracle Forms Server. To help you with this, the installer generates the dev6iconfig.txt file. This file contains details about the automatic configuration performed by the installer and the manual steps you must take to complete the process.

You'll find the dev6iconfig.txt file in %ORACLE_HOME%\orainst or $ORACLE_HOME/orainst. Once you finish installing Oracle Forms Server, read this file and perform the requested actions.

The following text is an example of the type of information that appears in the dev6iconfig.txt file. Items marked [action] are manual steps you must take to complete the configuration process. Items marked [info] are the automatic configuration steps that were completed by the installer.

```
This file gives details of the configuration steps done for you by the installation
process, marked [info], and any actions you need to do manually, marked [action].

Oracle WebDB listener message

[action] If you are using the Oracle WebDB listener and if the messages below state that
virtual paths have been created for you, you must restart the listener to have the changes
taken into account (this may be done from the Services tool on the Windows Control Panel).

Virtual path message (webdb)

[info] Virtual path <name> has been created with your WebDb listener, mapping to physical
path <physical_path>.

Virtual path message (not webdb)

[action] Please configure the following virtual path with your web listener: <vpname> for
physical directory <dir>

Service startup message (NT only)

[info] NT service <servicename> has been created and started for the <componentname>

Process startup message (UNIX only)

[action] A command to start the <componentname> has been written to the Oracle Forms
Developer process startup script <scriptfilename>.

Reports Server TNS message

[info]The following TNS name for the Reports Server has been defined in the Net8
<tnsnames.ora> file : <TNSname>

Reports Service message (NT only)

[action] NT service Oracle Reports Server [<servername>] has been created for the Reports
Multitier Server.  This must be started under an NT user with access to a printer before
any reports may run on this machine.

Reports CGI message
```

```
[action] Oracle Reports Web CGI and cartridge: in order to run reports on other machines,
your must add entries to your Net8 "tnsnames.ora" file to identify the Reports Servers
running on those machines.

Primary node install message

[action] You have requested that this machine be the primary node in a multiple machine
configuration.  If you wish to distribute the execution of Oracle Reports reports across
the different machines using Reports Server clustering, you will need to add the
CLUSTERCONFIG parameter to the Reports Server configuration file.  Please refer to the
Reports Server documentation for details.

Test form message

[info] An internet shortcut to run a standard test form, or any form of your choice, has
been created for you in the Oracle Forms Developer program group.

Test report message

[info] An internet shortcut to run a standard test report, or any report of your choice,
has been created for you in the Oracle Reports Developer program group.
```

STEP 5: CONFIGURING THE FORMSWEB.CFG FILE

The formsweb.cfg file is a powerful new convenience included with Oracle Forms Server 6*i*. Use it as a repository for all the settings you need to run Oracle forms on the Web in a CGI implementation. The installer places this file in %ORACLE_HOME%\FORMS60\server or $ORACLE_HOME/FORMS60/server.

This section:

- Describes the purpose and general organization of the formsweb.cfg file
- Lists and describes optional and mandatory parameters in the formsweb.cfg file
- Provides an example of a default formsweb.cfg file
- Discusses case sensitivity issues in the formsweb.cfg file

UNDERSTANDING THE FORMSWEB.CFG FILE

The formsweb.cfg file is a text file that contains configuration parameters for running Forms applications on the Web in a CGI implementation. The configuration parameters in the formsweb.cfg file are the CGI-equivalent of the cartridge parameters used with the Forms cartridge. The formsweb.cfg file is divided into three main sections:

- System Parameters
- User Parameters
- Specific Configurations

System Parameters

The System Parameters section provides information required by the Forms Web CGI. Unlike many other parameters in formsweb.cfg, System Parameters cannot be specified in a URL query string. However, you can override their values by placing an alternate parameter/value set in a Specific Configuration section in formsweb.cfg, then calling that configuration in the application URL.

User Parameters

The User Parameters section is where you specify the actual values for parameters that are defined with variables in the base HTML file. For example, in the base.htm file you might have:

<PARAM NAME="separateFrame" VALUE="%separateFrame%">

In the formsweb.cfg you would set the specific value for the variable %separateFrame%:

separateFrame=false

You can override specified User Parameter values in a Specific Configuration section in formsweb.cfg or in a query string in the application's URL. For example:

NT: http://servername.my.domain.name.com/dev60cgi/ifcgi60.exe?separateFrame=true
UNIX: http://servername.my.domain.name.com/dev60cgi/f60cgi?separateFrame=true

In these examples, the query string ?separateFrame=true will override the value for separateFrame that is specified in the formsweb.cfg file.

> *Note: When a specific value for a parameter is defined in both the formsweb.cfg file and the application's URL, the value defined in the URL is used.*

Specific Configurations

If you want to run the same form with multiple configurations, you can define custom configurations with custom values in the Specific Configurations section of the formsweb.cfg file. When you call the custom configuration with a query string in the application's URL, the custom values will override the parameters defined in the User Parameters section of formsweb.cfg. When you set up a Specific Configurations section, you need only specify the parameters you want to change. The default values that are specified in the User Parameters section will be used for all other parameters.

Use the "config" parameter in the application's URL to call a particular Specific Configuration section. For example, the following URLs call the Specific Configuration section [myconfig]:

NT: http://servername.my.domain.name.com/dev60cgi/ifcgi60.exe?config=myconfig
UNIX: http://servername.my.domain.name.com/dev60cgi/f60cgi?config=myconfig

FORMSWEB.CFG PARAMETERS

Most of the parameters in the formsweb.cfg file are set automatically as you select configuration options during Oracle Forms Server 6*i* installation. This section provides information on the parameters in the formsweb.cfg file, in the event that you wish to manually adjust configuration options.

formsweb.cfg Parameter	**Requirement**	**Definition**
baseHTML	Required	Use this parameter to specify the physical path to the HTML file that contains Forms applet tags. It allows the Forms applet to run under a Web browser's native Java Virtual Machine (JVM) and is suitable for use with the AppletViewer.
baseHTMLJInitiator	Optional	Use this parameter to specify the physical path to the base HTML file to be used if the user's browser is running on Windows and requires Oracle JInitiator to run the Forms applet. It should reference a file that contains tags with a syntax that is appropriate to JInitiator. When a value is not specified for the baseHTMLJInitiator parameter, the baseHTML parameter and value are used.

ie50	Optional	This parameter is provided for use in conjunction with the baseHTMLJInitiator parameter. It sets up conditional use of either JInitiator or the native JVM within the Internet Explorer 5.0 browser. Permitted values are: • JInitiator, which means the file referenced in the baseHTMLJInitiator parameter will be used. • Native, which means the Forms applet will run natively under IE5.0, without the help of JInitiator. The advantage of running native in an IE5.0 browser is the elimination of JInitiator download time.
HTMLdelimiter	Optional	Use this parameter to define the delimiter that is used to indicate variables in the base.htm and basejini.htm configuration files. The default is %. For example, you can indicate the variable value for connectMode with %connectMode%. Then you can define the value for the %connectMode% variable in the formsweb.cfg file or via a query string in the application's URL.
MetricsServerHost	Optional	This parameter is optional for Forms CGI (even when you use load balancing). If it is not supplied, the load balancer server must be running on the local machine—the same machine as the Web listener and the Forms CGI (ifcgi60.exe on NT, f60cgi on UNIX).
MetricsServerPort	Depends	This parameter is required for load balancing; otherwise it is optional. Use it to specify the port for contacting the load balancer server. It must match the Request Port value that you specify during installation (see page 9).
MetricsServerErrorURL	Optional	This parameter is optional for the Forms CGI (even when you use load balancing). Use it to specify the Web page that will be displayed if no valid %leastloadedhost% can be found. If it is not specified, or if its value is empty, the Forms CGI will automatically generate an error page with an appropriate message.
MetricsTimeout	Optional	It is used for load balancing. Use it to specify the time to wait for an answer from the load balancer server when getting the least loaded host. The default value is 30 seconds.
leastloadedhost	Depends	This parameter is required for load balancing; otherwise it is optional. This is a variable that can be specified in either the base HTML file or the formsweb.cfg file, wherever the name of the least loaded machine is required for load balancing. If you use the default base HTML file, which we recommend, then whenever you will use load balancing, be sure to specify the serverHost parameter in the formsweb.cfg file in the following way: serverHost=%leastloadedhost%

Standard HTML Applet Parameters

The following parameters are standard HTML applet parameters. Those that control aspects of display, such as *align* and *valign*, are not used when the applet is displayed in a separate browser window (i.e., separateFrame=true). When the applet is displayed in a separate browser window, default values control the display.

In the base HTML file, you can define a parameter with either a value or a variable. If you specify a variable (recommended), put the actual value in the formsweb.cfg file or specify a value via a query string in the application's URL. For example, in the base HTML file you can specify:

<PARAM NAME="width" VALUE="%width%">

Then you can define the value for width in the formsweb.cfg file or in a query string in the application's URL

For example:

formsweb.cfg file: width=500

Application URL NT: http://servername.my.domain.com/dev60cgi/ifcgi60.exe?width=500

Application URL UNIX: http://servername.my.domain.com/dev60cgi/f60cgi?width=500

The syntax for parameters varies slightly between the base.htm and basejini.htm files. See pages 23 and 24 for examples of parameter use in these files.

codebase	Required	Use this parameter to list the virtual directory you defined in your Web server to point to the physical directory where Java files are stored. (See page 12.) Any paths given in the *archive* and *code* parameters are relative to this directory (URL). Although it's possible to include this parameter and value in a query string in the application's URL, we advise that you do not.
code	Required	This parameter has a fixed value. Its value should always be: oracle.forms.engine.Main. Although it's possible to include this parameter and value in a query string in the application's URL, we advise that you do not.
archive	Optional	Use this parameter to specify a comma-separated list of archive (JAR) files to preload. Paths are relative to the value defined in *codebase*, or they are absolute paths. Although it's possible to include this parameter and value in a query string in the application's URL, we advise that you do not.
width	Required	Use this parameter to specify the width of the form. Express the value in pixels.
height	Required	Use this parameter to specify the height of the form. Express the value in pixels.
align	Optional	Use this parameter to specify the horizontal position of the forms applet window relative to computer screen display. Accepted values are left, right, center.
valign	Optional	Use this parameter to specify the vertical position of the forms applet window relative to computer screen display. Accepted values are top, middle, bottom.
alt	Optional	Use this parameter to define the message text that displays in lieu of the applet, when applets are not supported in the client browser.

hspace	Optional	Use this parameter to set the horizontal gutter between the applet and the applet window frame. Values are measured in pixels. By default, hspace is a small, non-zero number.
vspace	Optional	Use this parameter to set the vertical gutter between the applet and the applet window frame. Values are measured in pixels. By default, vspace is a small, non-zero number.
type	Depends	This parameter is mandatory for JInitiator. Its fixed value is: application/x-jinit-applet No value is required for AppletViewer.
name	Optional	Use this parameter to list the Applet instance name.
title	Optional	Use this parameter to list the advisory title string.
border	Optional	Use this parameter to specify the width of the outer border around the applet to a given number of pixels (e.g., border=4). To suppress the border, set its value to zero or omit the parameter.
standby	Optional	Use this parameter to define the message text that displays when the applet is loading.
codetype	Optional	This parameter defaults to the value specified for *type*.

Forms Applet Parameters

The following parameters are specific to the Forms applet. This means they are specific to Oracle, rather than standard to HTML. Specify values in the base HTML file, or specify a variable in the base HTML file, then put the actual value in the formsweb.cfg file or include it as a query string in the application's URL. For example, in the base HTML file you can specify:

<PARAM NAME="connectMode" Value="%connectMode%">

Then you can define the value for connectMode in the formsweb.cfg file or include it as a query string in the application's URL. For example:

formsweb.cfg file: connectMode=https

Application URL NT: http://servername.my.domain.com/dev60cgi/ifcgi60.exe?connectMode=https

Application URL UNIX: http://servername.my.domain.com/dev60cgi/f60cgi?connectMode=https

The syntax for parameters varies slightly between the base.htm and basejini.htm files. See pages 23 and 24 for examples of parameter use in these files.

serverHost	Depends	This parameter is mandatory when you implement load balancing; otherwise, it is optional. With load balancing, specify this parameter value as follows in the formsweb.cfg file: serverHost=%leastloadedhost% This parameter specifies the host for the Forms server (ifsrv60.exe on NT; f60srvm on UNIX). In the presence of load balancing, it always selects the least loaded host. In the absence of load balancing, it defaults to the machine that hosts the Web server.
serverPort	Required	Use this parameter to specify the port on which the Forms server (ifsrv60.exe on NT; f60srvm on UNIX) listens. In most cases, the value will be 9000, which is also the default

serverArgs	Required	Use this parameter to specify the command line parameters for Runform (see module, userid, and user-defined parameters, below).
heartBeat	Optional	Use this parameter to set the frequency at which a client sends a packet to the server to indicate that it is still running. Define this value in minutes. The default is two minutes.
imageBase	Optional	Use this parameter to indicate where icon files are stored. Choose between: • codeBase indicates that the icon search path is relative to the directory that contains the Java classes. Use this value if you store your icons in a JAR file (recommended). • documentBase is the default. In deployments that make use of the Forms Server CGI, you must specify the icon path in a custom application file. Refer to Forms Server documentation for further information.
registryPath	Optional	Use this parameter to list the virtual directory where the application file named in the serverApp parameter is located. For information on relevant virtual directories, see page 12.
webformsTitle	Optional	Use this parameter to change the title that appears in the top border of a form's display window.
splashScreen	Optional	Use this parameter to specify the GIF or JPG file that will appear before the applet appears. Set the value to “no” for no splash screen. Leave it empty to use the default splash screen supplied by Oracle.
background	Optional	Use this parameter to specify the GIF or JPG file that should appear in the background. Set the value to “no” for no background image. Leave it empty to use the default background image that is supplied by Oracle.
connectMode	Depends	This parameter is mandatory for HTTP and HTTPS connections. It is optional for socket connections. Use it to tell the client which type of connection protocol to use with the Forms Server. Choose from: socket, http, https. The default is *socket.*
clientDPI	Optional	Use this parameter to manage varying dots-per-inch (DPI) settings per platform. For example, a form developed on the Win32 platform may not display properly on the UNIX platform due to varying DPI values. The client DPI value can be any positive integer. We recommend that you use an integer between 50 and 200. This parameter's value overrides the value defined in the JVM
separateFrame	Optional	Use this parameter to specify whether the applet opens in the primary (false) or a secondary (true) browser window. If you choose to open the applet in a secondary browser window, any customized, non-default appearance parameters you define in formsweb.cfg or a base HTML file will be ignored. Default parameters will control the look of the secondary window.

lookAndFeel	Optional	Use this parameter to define the application’s look and feel. Valid values are Oracle or Generic. Generic invokes a Windows 95 look and feel.
colorScheme	Optional	Use this parameter to define the application’s color scheme. Valid values are Teal, Titanium, Red, Khaki, Blue, Olive, or Purple. The colorScheme parameter is ignored if lookAndFeel is set to Generic.
serverApp	Optional	Use this parameter to specify the name of your application class (if any). Application classes are used for creating application-specific font mapping and icon path settings.
module	Required	This parameter is a subparameter of the serverArgs parameter. Use it to specify the form module name and, if necessary, the module’s path. It is expressed as: <PARAM NAME=“serverArgs” VALUE=“module=order.fmx”>
userid	Optional	This parameter is a subparameter of the serverArgs parameter. Use it to specify a user login string, such as scott/tiger@ORA8. It is expressed as: <PARAM NAME=“serverArgs” VALUE=“userid=scott/tiger@ORA8”>
user-defined parameters	Optional	Use these parameters to specify user-defined name/value pairs. For example: <PARAM NAME=“name” VALUE=“value”>

DEFAULT FORMSWEB.CFG FILE

This section provides an example of the formsweb.cfg file. The values that appear in your version of this file will vary according to the choices you make when you install Oracle Forms Server. As part of its installation, Oracle Fomrs Server automatically generates the formsweb.cfg file and places it in %ORACLE_HOME%\FORMS60\server or $ORACLE_HOME/FORMS60/server.

```
; Forms Web CGI Configuration File
; This file defines parameter values used by the Forms Web CGI
; ********************************
; PARAMETER VALUES USED BY DEFAULT
; ********************************

  ; SYSTEM PARAMETERS
  ; -----------------
  ; These have fixed names and give information required by the Forms
  ; Web CGI in order to function.  They cannot be specified in the URL
  ; query string.  But they can be overridden in a named configuration
  ; (see below).
baseHTML=%ORACLE_HOME%\FORMS60\server\base.htm
baseHTMLJInitiator=%ORACLE_HOME%\FORMS60\server\basejini.htm
HTMLdelimiter=%
MetricsServerPort=9020
MetricsServerErrorURL=
  ; The next parameter specifies how to execute the Forms applet under
  ; Microsoft Internet Explorer 5.0.  Put IE50=native if you want the
  ; Forms applet to run in the browser’s native JVM.
IE50=JInitiator
```

```
  ; USER PARAMETERS
  ; ---------------
  ; These match variables (e.g. %form%) in the baseHTML file. Their
  ; values may be overridden by specifying them in the URL query string
  ; (e.g."http://myhost.mydomain.com/ifcgi60.exe?form=myform&width=700")
  ; or by overriding them in a specific, named configuration (see below)

  ; 1) Runform arguments:
form=test.fmx
userid=
otherparams=

  ; 2) HTML page title, attributes for the BODY tag, and HTML to add
  ; before and after the form:
pageTitle=Oracle Forms Server
HTMLbodyAttrs=
HTMLbeforeForm=
HTMLafterForm=

  ; 3) Values for the Forms applet parameters:
width=650
height=500
separateFrame=false
splashScreen=no
background=no
lookAndFeel=Oracle
colorScheme=teal
serverApp=default
serverPort=9000
serverHost=
connectMode=socket
archive=f60web.jar

  ; 4) Parameters for JInitiator
    ; Page displayed to Netscape users to allow them to download
    ; JInitiator. If you create your own version, set this parameter to
    ; point to it.
jinit_download_page=/jinitiator/us/jinit_download.htm
    ; Parameters related to the version of JInitiator.
    ; These are valid for Oracle JInitiator version 1.1.7.21o
    ; WARNING: You must update these if you upgrade to a later version
    ; of JInitiator (as instructed in the documentation for that
    ; version)
jinit_classid=clsid:b8b42d42-6bba-11d3-a3c3-00c04fa32518
jinit_exename=jinit.exe#Version=1.1.7.21
jinit_mimetype=application/x-jinit-applet;version=1.1.7.21

; ********************************
; SPECIFIC CONFIGURATIONS
; ********************************
; You may define your own specific, named configurations (sets of
; parameters) by adding special sections as illustrated in the following
; examples. Note that you need only specify the parameters you want to
; change. The default values (defined above) will be used for all other
; parameters. Use of a specific configuration can be requested by
; including the text "config=<your_config_name>" in the query string of
; the URL used to run a form.  For example, to use the sepwin
; configuration, your could issue a URL like
; "http://myhost.mydomain.com/ifcgi60.exe?config=sepwin".

; Example 1: configuration to run forms in a separate browser window
; with "generic" look and feel (include "config=sepwin" in the URL)
[sepwin]
separateWindow=True
lookandfeel=Generic

; Example 2: configuration affecting users of MicroSoft Internet
; Explorer 5.0. Forms applet will run under the browser’s native JVM
; rather than using Oracle JInitiator.
[ie50native]
IE50=native
```

```
; Example 3: configuration forcing use of the base.htm base HTML file in
; all cases (means applet-style tags will always be generated and
; JInitiator will never be used).
[applet]
baseHTMLJInitiator=
```

CASE SENSITIVITY IN FORMSWEB.CFG

In the past, some application deployments have been hindered by the incorrect use of upper- and lowercase letters in configuration parameters and filenames. Wherever possible, we are eliminating case requirements. For example, most parameters in the formsweb.cfg file are not case sensitive. Nonetheless, you must use caution when specifying Forms applet tag parameters, which are case sensitive, whether in the formsweb.cfg file or in one of the base HTML files. Additionally, the case sensitivity of directory and file names varies according to the requirements of your operating system.

For the time being, a good rule of thumb is to use the cases for parameter names and values presented in the following sections and to always adhere to the requirements of your operating system when specifying the cases of file and directory names.

STEP 6: CONFIGURING THE BASE.HTM OR BASEJINI.HTM FILE

When you start a Web-enabled application (by clicking a link to the application's URL), the Forms CGI reads a special file that contains all necessary applet tags, parameters, and parameter values (or variables for those values) that are required to run the selected application on the Web. This is the *base HTML* file.

The Oracle Forms Server 6*i* installer places two base HTML files in the following directory %ORACLE_HOME%\FORMS60\server (NT) or $ORACLE_HOME/FORMS60/server (UNIX):

- basejini.htm
  This file contains the tags required to run the Forms applet using a combination of the user's Web browser and Oracle JInitiator.
- base.htm
  This file contains the tags required to run the Forms applet in the AppletViewer or in any Web browser certified by Oracle whose native JVM is certified to work with Forms.

In a Forms Web CGI implementation, as the application launch process gets started, any variables (%*variablename*%) in the base HTML file are replaced with the appropriate parameter values that are specified either in the formsweb.cfg file or in a query string included in the application's URL. Once all values are defined, the HTML file is generated and then downloaded to the user's Web browser, and the selected forms application launches.

> *Note: When a specific value for a parameter is defined in both the formsweb.cfg file and the application's URL, the value defined in the URL is used.*

In most cases, you will not need to modify the default base HTML files. Instead, you can define their parameters with variables. Then you can define the actual values for the variables in formsweb.cfg or in the application's URL.

For example, you can define the parameter splashScreen in the base HTML file as:

<PARAM NAME="splashScreen" VALUE="%splashScreen%">

Then define the actual value in the formsweb.cfg file as:

NT: splashScreen=*directory path*\mysplashscreen.gif
UNIX: splashScreen=*directory path*/mysplashscreen.gif

If you prefer, you can define the splashScreen variable in the application's URL:

NT: http://servername.my.domain.com/dev60cgi/ifcgi60.exe?splashScreen=*directory path*/mysplashscreen.gif

UNIX: http://servername.my.domain.com/dev60cgi/f60cgi?splashScreen=*directory path*/mysplashscreen.gif

> *Note: For your splash screen, you can use your own graphic or the default file that Oracle provides.*

Using variables instead of values in the base HTML file allows you to use the same generic base HTML file for all your forms applications and to manage configuration complexity from one location: formsweb.cfg (or the application's URL).

> *Note: If you decide to specify parameter values in the base HTML file, do not modify the original base HTML file that is provided by Oracle. Instead, modify a renamed copy. Be sure to update the baseHTML (or the basejiniHTML) parameter in the formsweb.cfg file to point to the location of the modified file.*

DEFAULT BASE.HTM FILE

```
<HTML>
<!-- FILE: base.htm (Oracle Forms Developer)                        -->
<!-- This is the default base HTML file for running a form on the   -->
<!-- web using APPLET-style tags to include the Forms applet.       -->
<!-- This file will be REPLACED if you reinstall "Forms Web CGI and -->
<!-- cartridge", so you are advised to make your own version if you -->
<!-- want to make any modifications.  You should then set the       -->
<!-- baseHTML parameter in the Forms web CGI configuration file     -->
<!-- (formsweb.cfg) to point to your new file instead of this one.  -->

<!-- IMPORTANT NOTE: default values for all the variables which     -->
<!-- appear below (delimited by the percent character) are defined  -->
<!-- in the formsweb.cfg file. It is preferable to make changes in  -->
<!-- that file where possible, and leave this one untouched.        -->

<HEAD><TITLE>%pageTitle%</TITLE></HEAD>

<BODY %HTMLbodyAttrs%>
%HTMLbeforeForm%

<!-- Forms applet definition (start) -->

<APPLET CODEBASE="/forms60java/"
         CODE="oracle.forms.engine.Main"
         ARCHIVE="%archive%"
         WIDTH="%Width%"
         HEIGHT="%Height%">

<PARAM NAME="serverPort" VALUE="%serverPort%">
<PARAM NAME="serverHost" VALUE="%serverHost%">
<PARAM NAME="connectMode" VALUE="%connectMode%">
<PARAM NAME="serverArgs" VALUE="module=%form% userid=%userid% %otherParams%">
<PARAM NAME="separateFrame" VALUE="%separateFrame%">
<PARAM NAME="splashScreen"  VALUE="%splashScreen%">
<PARAM NAME="background"  VALUE="%background%">
<PARAM NAME="lookAndFeel"  VALUE="%lookAndFeel%">
<PARAM NAME="colorScheme"  VALUE="%colorScheme%">
<PARAM NAME="serverApp" VALUE="%serverApp%">
</APPLET>
```

```
<!-- Forms applet definition (end) -->

%HTMLafterForm%

</BODY>
</HTML>
```

Usage Notes

- If you do not use a parameter tag in the default base.htm file, delete it from the file.
- All variables must receive values at runtime, either from the formsweb.cfg file or within a query string in the application's URL. If a variable does not receive a value, the Forms Server cannot generate an HTML file to pass back to the user's Web browser. This results in an error that takes you to a Web page that lists the undefined parameter.

DEFAULT BASEJINI.HTM FILE

```
<HTML>
<!-- FILE: basejini.htm (Oracle Forms Developer)                  -->
<!-- This is the default base HTML file for running a form on the   -->
<!-- web using JInitiator-style tags to include the Forms applet.   -->
<!-- This file will be REPLACED if you reinstall "Forms Web CGI and -->
<!-- cartridge", so you are advised to make your own version if you -->
<!-- want to make any modifications.  You should then set the       -->
<!-- baseHTML parameter in the Forms web CGI configuration file     -->
<!-- (formsweb.cfg) to point to your new file instead of this one.  -->

<!-- IMPORTANT NOTE: default values for all the variables which     -->
<!-- appear below (delimited by the percent character) are defined  -->
<!-- in the formsweb.cfg file. It is preferable to make changes in  -->
<!-- that file where possible, and leave this one untouched.        -->

<HEAD><TITLE>%pageTitle%</TITLE></HEAD>

<BODY %HTMLbodyAttrs%>
%HTMLbeforeForm%

<!-- Forms applet definition (start) -->
<OBJECT classid="%jinit_classid%"
         codebase="/jinitiator/%jinit_exename%"
         WIDTH="%Width%"
         HEIGHT="%Height%"
         HSPACE="0"
         VSPACE="0">

<PARAM NAME="TYPE"       VALUE="%jinit_mimetype%">
<PARAM NAME="CODEBASE"   VALUE="/forms60java/">
<PARAM NAME="CODE"       VALUE="oracle.forms.engine.Main" >
<PARAM NAME="ARCHIVE"    VALUE="%archive%" >
<PARAM NAME="serverPort" VALUE="%serverPort%">
<PARAM NAME="serverHost" VALUE="%serverHost%">
<PARAM NAME="connectMode" VALUE="%connectMode%">
<PARAM NAME="serverArgs" VALUE="module=%form% userid=%userid% %otherParams%">
<PARAM NAME="separateFrame" VALUE="%separateFrame%">
<PARAM NAME="splashScreen"  VALUE="%splashScreen%">
<PARAM NAME="background"  VALUE="%background%">
<PARAM NAME="lookAndFeel"  VALUE="%lookAndFeel%">
<PARAM NAME="colorScheme"  VALUE="%colorScheme%">
<PARAM NAME="serverApp" VALUE="%serverApp%">
<COMMENT>
<EMBED SRC="" PLUGINSPAGE="%jinit_download_page%"
         TYPE="%jinit_mimetype%"
         java_codebase="/forms60java/"
         java_code="oracle.forms.engine.Main"
         java_archive="%archive%"
         WIDTH="%Width%"
```

```
          HEIGHT="%Height%"
          HSPACE="0"
          VSPACE="0"

          serverPort="%serverPort%"
          serverHost="%serverHost%"
          connectMode="%connectMode%"
          serverArgs="module=%form% userid=%userid% %otherparams%"
          separateFrame="%separateFrame%"
          splashScreen="%splashScreen%"
          background="%background%"
          lookAndFeel="%lookAndFeel%"
          colorScheme="%colorScheme%"
          serverApp="%serverApp%"
>
<NOEMBED>
</COMMENT>
</NOEMBED></EMBED>
</OBJECT>
<!-- Forms applet definition (end) -->

%HTMLafterForm%
</BODY>
</HTML>
```

Usage Notes

- If you do not use a parameter tag in the basejini.htm file, delete it from the file.
- All variables must receive values at runtime, either from the formsweb.cfg file or within a query string in the application's URL. If a variable does not receive a value, the Forms Server cannot generate an HTML file to pass back to the user's Web browser. This results in an error that takes you to a Web page that lists the undefined parameter.
- To streamline performance, use only one Web server as a source for JAR file downloads. This will prevent multiple downloads of the same files from different servers.

STEP 7: BROADCASTING THE APPLICATION'S URL

To broadcast the application's URL, simply notify your intended users. Your users can contact the URL with their Java-enabled Web browsers and run the corresponding application. For example, to announce the availability of its new Order Tracking application, ABC Corp. might notify employees via e-mail of the following URL:

NT: http://servername.my.domain.name.com/dev60cgi/ifcgi60.exe?config=myconfig&form=
tracker.fmx
UNIX: http://servername.my.domain.name.com/dev60cgi/f60cgi?config=myconfig&form=
tracker.fmx

ABC's URL consists of the following components:

http	Connection protocol
servername	Name of the machine that hosts the application server
my.domain.name.com	Name of the domain that hosts the target information

dev60cgi	The virtual path, defined in the Web server, that points to CGI executables (scripts)
ifcgi60.exe (NT) f60cgi (UNIX)	Forms CGI, which is used for load balancing.
?config=myconfig&form=tracker.fmx	The query string that points to a custom configuration defined in the user-created "myconfig" section of the formsweb.cfg file and to the form module tracker.fmx The parameter "form" is used here because "form" was defined as the variable value for "module" in the base HTML file. That is: <PARAM NAME="serverArgs" VALUE="module=%form%"> The syntax is slightly different in the base HTML JInitiator file: serverArgs="module=%form%"

SUMMARY

In this paper we have provided information about reconfiguring forms cartridge implementations to CGI. We've kept to a fairly narrow path of configuration options to ensure a smooth and successful migration. In doing so, we've barely scratched the surface of the tremendous flexibility available to you for configuring your environment when you deploy Oracle Forms Developer applications with Oracle Forms Server and (optimally) the WebDB listener in a CGI implementation.

For more information about deploying Web applications with Oracle Forms Server, see *Oracle Forms Developer: Deploying Forms Applications to the Web with Oracle Forms Server 6i*, available in the documentation set that accompanies Oracle Forms Developer 6*i* (Oracle Forms Developer Online Manuals). You can purchase a printed edition of this book on the Web at www.oracle.com. Follow the link to the Oracle Store, and select the Documentation link on the Tools tab.

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

Worldwide Inquiries:
+1.650.506.7000
Fax +1.650.506.7200
http://www.oracle.com/

This document is provided for informational purposes only, and the information herein is subject to change without notice. Please report any errors herein to Oracle Corporation. Oracle Corporation does not provide any warranties covering and specifically disclaims any liability in connection with this document.

Oracle is a registered trademark, and Oracle8*i*, Oracle8, PL/SQL, and Oracle Expert are trademarks of Oracle Corporation. All other company and product names mentioned are used for identification purposes only and may be trademarks of their respective owners.