Oracle Forms 10g – New Features

An Oracle Whitepaper
March 2004

ORACLE

Oracle Forms 10g – New Features

Oracle Forms 10g – New Features

INTRODUCTION

If you are running Oracle Forms6*i* today and need to know about all the new features added since then, please read the *Oracle9i Forms New Features* whitepaper available at otn.oracle.com/products/forms.

This paper focuses on the new features that have been added to Oracle Forms in Oracle Application Server 10g and Oracle Developer Suite 10*g*. Each new feature is explained in brief, with an outline of the feature and how it works.

This paper does not discuss the Forms Services architecture and therefore assumes that the reader is familiar with running Forms on the Web. To learn about the Forms Services Web architecture, please refer to the *Oracle9i Application Server Release 2, Forms Services Overview* whitepaper[1].

HOW ORACLE FORMS 10g IS DIFFERENT FROM ORACLE9*i* FORMS

Moving from Oracle9i Forms to Oracle Forms 10g is not the quantum leap you may have expected, given the version change from 9*i* to 10g. The change in version number follows the name change for the new release of Oracle Application Server, which Oracle Application Server 10*g* Forms Services is a component of. All samples and collateral provided for Oracle9*i* Forms are still relevant in Oracle Forms 10g.

The official Forms component names are “Oracle Application Server 10*g* Forms Services” for the Web runtime environment contained in Oracle Application Server 10g and "Oracle Forms Developer" for the application building environment, which is a part of Oracle Developer Suite 10g.

Though the Forms Services and Forms Developer names now contain the designator “10g,” the internal release version of Forms is Forms 9.0.4. Oracle Forms 10g is based on the same required support files (RSF) for the database as is Oracle9*i* Forms.

The 9.0.4 internal release number determines the names used for Forms components and system variables, like the Forms Servlet (f90servlet), the Forms Listener Servlet (l90servlet), the Forms Web context root (/ forms90/), the Forms Web runtime (ifweb90), the Forms path environment variable (Forms90_Path), - and so on.

Oracle Forms 10g does introduce some useful new functionality, and those features are discussed in this paper.

---

[1] http:/ / otn.oracle.com/ products/ forms/ pdf/ forms9iarchitecture.pdf

NEW FEATURES IN FORMS SERVICES

The following functionality has been added to Oracle Application Server 10*g* Forms Services.

Forms Services and Reports Services–only installation

Within Oracle Application Server 10*g*, you can now choose to install only those Oracle Application Server middle-tier components required by Oracle Application Server 10*g* Forms Services and Oracle Application Server 10*g* Reports Services. Customers who want to run Forms and Reports only on the Web will no longer be asked to install the application server infrastructure components.

The Forms and Reports–only installation option is available as a separate installation CD. It is not yet available as a menu option in the standard Oracle Application Server 10*g* installation.

The Forms and Reports–only installation includes the core J2EE features of Oracle Application Server 10*g*.

Not installing the Oracle Application Server 10*g* infrastructure, however, means that some functionality of the Application Server, such as single sign-on, is not available. Oracle Application Server 10*g* licensing also applies to the Forms and Reports–only installation option.

Runtime pre-start

The Forms runtime pre-start functionality allows startup of a configurable number of application runtime engines prior to their usage. This feature is designed to support quick connections at peak times and is used to shorten the server-side application startup time.

To enable an application to use this feature, configure the formsweb.cfg file for that application as described in this section before starting Oracle Application Server 10*g*. Because you can configure pre-start on an application by application basis, the feature allows you to set the relative weighing of each application running in the same Forms Services instance.

Set the following parameters in the formsweb.cfg file to enable and configure runtime pre-start for a Forms application

prestartRuntimes – The switch to enable or disable the runtime pre-start for an application. The default value of this parameter is *false,* indicating that no runtimes are started when - Oracle Application Server is started.

prestartInit – Defines the number of Forms runtime engines that are initially started for an application.

prestartMin – Defines the minimum number of pre-started runtimes that must exist at any time for a specific application. The minimum value must be less than or equal to what's defined for the *prestartInit* parameter. The *prestartMin* parameter can be modified at any time and does not require - the Application Server to be restarted.

prestartIncrement – Defines the additional engines to be started if the number of idle pre-started runtime engines for a specific application goes below the number of engines defined by the *prestartMin* parameterThe *prestartIncrement* parameter can be modified at any time and does not require an application server restart.

prestartTimeOut – Time specified in minutes indicating when pre-started runtimes should be shut down if not used. Because Forms runtime engines are pre-stared during application startup time, this value should be reasonable high for the runtime pre-start to have an effect. This parameter becomes more useful when runtime pre-start scheduling gets introduced in one of the upcoming Forms releases.

The following example enables runtime pre-start for the "myHR" application configuration in the formsweb.cfg file.

```
[myHR]

forms=hr_starter.fmx
lookandfeel=oracle
prestartRuntimes=true
prestartInit = 100
prestartMin = 50
prestartIncrement = 15
prestartTimeout = 2000
…
```

In the example configuration above, starting Oracle Application Server 10*g* starts 100 initial Forms runtime engines on this Forms Server instance. The minimum number of idle runtime engines is specified as 50. Each time this threshold is passed, 15 new engines are spawned.

A note of caution:

Overusing this feature may lead to performance issues if the middle tier server does not have enough memory to handle all prestarted runtime engines. Every Forms runtime engine running in idle mode will consume memory on the middle tier server.

URL Security

Oracle Application Server 10*g* Forms Services can be passed command line arguments either using the formsweb.cfg configuration file or parameters added to the request URL.

By design, command line arguments passed in a URL will always override similar definitions in the formsweb.cfg file. For example, an application that is configured

to connect to the database as scott/ tiger can connect as a different user if the userid parameter is added as a URL parameter.

URL security is introduced in Oracle Application Server 10*g* Forms Services to avoid misuse of URL parameters. The Forms administrator decides whether URL security should be applied to an application by setting the *restrictedURLparams* parameter in the formsweb.cfg file.

RestrictedURLparams – A comma separated list of Forms URL parameters that are blocked from being added directly to a Forms Services request URL. The RestrictedURLparams parameter can be defined in the default configuration section of the formsweb.cfg file, or individually for an application in its named configuration section.

The following example prevents users from adding the "pageTitle", "width", "height", and "otherparams" parameters directly to the request URL when starting the myHR Forms App.

```
[myHR]

forms=hr_starter.fmx
lookandfeel=oracle
RestrictedURLparams=pageTitle,width,height,otherparams
```

A user that tries to append a URL parameter contained in the list of restricted parameter will receive an error message from the system.

Improved single sign-on functionality

Oracle9*i*AS Forms Services running in Oracle9*i* Application Server already was able to leverage Oracle single sign-on without any change required in the Forms application modules.

The Forms single sign-on configuration in Oracle9*i* Application Server was static, in that either all applications served by a Forms Services instance run in single sign-on mode, or none. Forms Services in Oracle Application Server 10g offers a more dynamic and flexible configuration for Forms applications running in single sign-on mode. Error handling is greatly improved and allows customization.

Dynamic directives

An in depth introduction to single sign-on in Oracle Application Server 10*g* Forms Services is available in the Whitepaper "Oracle Application Server 10*g* (9.0.4) - Forms Single Sign-On

With Oracle Application Server 10*g* Forms Services, the configuration of single sign-on is done win the formsweb.cfg configuration file, located in the forms90/ server subdirectory. Using the formsweb.cfg file, instead of the forms90.conf file that was used with Oracle9iAS Forms Services, allows customers

to run single sign-on protected applications and publicly accessible applications on the same Forms Services instance.

ssoMode is a new configuration parameter in the formsweb.cfg file that, if set to true, enforces single sign-on for individual applications, if set in a named configuration section, or all applications, if set in the user parameter section.

ssoMode = false| true

The following example enforces single sign-on authentication to be used for the myHR application:

```
[myHR]

forms=hr_starter.fmx
lookandfeel=oracle
ssoMode = true
…
```

Handling of a missing Resource Access Descriptor (RAD)

Oracle Forms requires a database username and password pair to connect an application to the database using SQL*Net.

Forms applications running in single sign-on mode obtain the single sign-on user's database credentials through the Forms Servlet. Upon application request, the Forms Servlet reads the database connect information from the Oracle Internet Directory (OID), a LDAP v3 compliant directory server that is in the core of the single sign-on solution in Oracle Application Server 10g.

An in depth introduction to single sign-on in Oracle Application Server 10g Forms Services is available in the Whitepaper "Oracle Application Server 10*g* (9.0.4) - Forms Single Sign-On

The Forms application database credentials are stored in Resource Access Descriptors (RAD)[2], which are special entries in OID that are located by the Forms Servlet through a unique key built out of the user's single sign-on name and the value provided as the URL 'config' parameter.

SsoDynamicResourceCreate – set to true or false to determine if a missing RAD resource in Oracle Internet Directory shall be created by Forms on behalf of the application user. To provide the database access information for this application, an HTML form will be presented to the user. The RAD resource created contains the name specified as the value of the config parameter in the request URL. The default value of this parameter is "true".

SsoErrorURL – If you don't want Forms to dynamically create missing RAD entries in OID then set the SsoDynamicResource parameter to false and use this parameter to specify a URL that a user gets directed to. This parameter is useful if

2 Please refer to the chapter 28 "Oracle Delegated Administration Services" of the "Oracle Internet Directory Administrator's Guide" for how to create Resource Access Descriptors for Forms and Reports

you want to handle the situation where a RAD entry cannot be found based on the user misspelling the value of the config parameter.

SsoCancelURL – If setting SsoDynamicResourceCreate to true, the user is presented a HTML form to create a new RAD entry whenever this entry is missing for the application he tries to run. If the user presses the cancel button of this form then the SsoCancelURL parameter can be used to direct the user to a custom help page or any other URL.

Support for default Resource Access Descriptors

One configuration option of the Delegated Administration Service (DAS) is to specify default Resource Access Descriptors. Default RAD entries are accessible for every single sign-on user authenticated in OID, which means that all users will use the same database account to connect to the database.

Supporting default resource access descriptors makes configuring single sign-on easier to use for Forms Services administrators because they don't need to create Resource Access descriptors for each individual user.

Handling password expiry

Oracle Application Server 10*g* Forms Services store database usernames and password combinations in Oracle Internet Directory (OID) using Resource Access Descriptors (RAD). Resource Access Descriptors are LDAP entries that securely store a user's database credentials for a particular Forms application running in single sign-on mode.

A Forms functionality added in Oracle Forms6i is the ability to re-new a database password through the running Forms application if it has expired. Oracle Application Server 10*g* Forms Services now updates the RAD entry in OID that holds the database connect information for the authenticated user and the current application. Note that RAD entries are only updated for Forms applications that run in single sign-on mode.

New GET_APPLICATION()_PROPERTY arguments for sso

In Oracle9*i*AS, using the Forms GET_APPLICATION_PROPERTY() built-in application developers were able to obtain the lightweight username of the single sign-on authenticated application user.

In Oracle Forms 10g, two additional single sign-on attributes are exposed to the Forms application developer during runtime:

1. *Distinguished Username* – A single sign-on username 'foo' may be unique within branch of the LDAP directory tree (Oracle Internet Directory), but not necessarily in the whole directory.

   The distinguished name is unique in a LDAP tree and consists of the single sign-on username and its complete access path in the directory tree.

For example, the username 'foo' could be located in the 'us' branch and the 'de' branch of the Oracle Internet Directory. In this case the unique distinguished names are :
`cn=foo,cn=users,dc=us,dc= oracle,dc=com` and
`cn=foo,cn=users,dc=de,dc=oracle,dc=com.`

To obtain the user distinguished name in Forms, the GET_ APPLICATION_ PROPERTY() built-in expects the argument 'sso_usrdn' to be passed.

Subscriber Name – The subscriber name determines the branch of the LDAP tree that the username is located in. The subscriber name for the user DN

`cn=foo,cn=users,dc=us,dc=oracle,dc=com`

is

`dc=us,dc=oracle,dc=com`

To obtain the subscriber name of a user entry in Oracle Internet Directory, the GET_APPLICATION_ PROPERTY() built-in expects the argument 'sso_subdn' to be passed.

Enhanced Oracle Enterprise Manager support

OracleEnterprise Manager 10g Application Server Control (OEM) can be used to configure and monitor Oracle Application Server 10*g* Forms Services applications.

The Forms Services configuration page in OracleEnterprise Manager 10g Application Server Control can be accessed from the midtier link. In a default Oracle Application Server 10g installation, the Oracle Enterprise Manager 10g Application Server Control page for the midtier is accessible through port 1815.

On selecting the "Forms" component from the list of components on the main OEM page of the middle-tier, a number of different tab pages are displayed

- Overview
- User Sessions
- Configuration
- Environment
- Forms Utility

The Oracle Application Server 10*g* Forms Services overview tab in Oracle Enterprise Manager 10g Application Server Control provides general information about the state of the Forms Services and the CPU and memory usage Forms has on this server instance (Figure 1).

A new feature added to Forms Services in Oracle Application Server 10g is the ability to allow or disallow new connection request to be handled by the Forms Servlet.

The green traffic light image that is shown on the overview tab page means that the Forms Servlet handles incoming user application requests. On clicking the "Disable" button, the traffic light changes to yellow, which means that only those users that are currently connected can work on this particular Forms Services instance.

This feature is useful where administrative work is scheduled which requires no user to be connected to a Forms Services application. Instead of killing current user processes, the “New Connections Enabled” feature can be used to first stop Forms Services from handling new incoming application requests and wait for working user processes to stop.

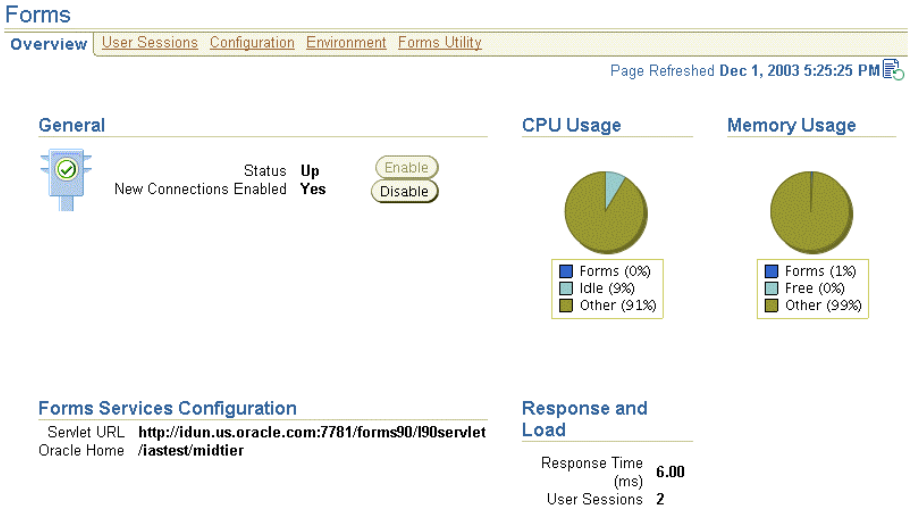

Figure 1: Oracle Application Server 10*g* Forms Servicves OEM Overview page

Technically, pressing the "Disable" button adds a new parameter "allowNew Connections" with its value set to “false” to the default section of the formsweb.cfg file. New application requests will get redirected to a Web URL identified by the "connectionDisallowedURL" parameter, which also is new in Oracle Application Server 10*g* Forms Services.

The following message is shown if the “connectionDisallowedURL” is not set while new application connects are disabled:

"The Forms10g Servlet will not allow new connections. Please contact your System Administrator."

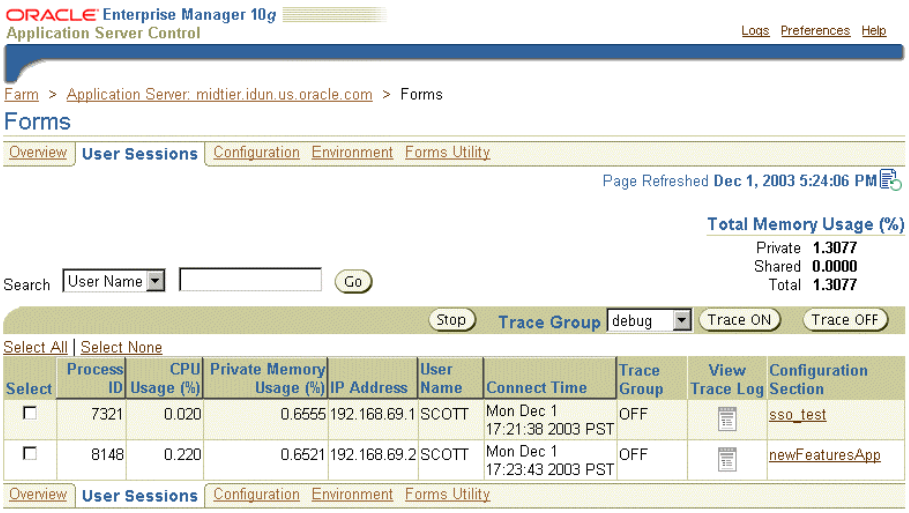

Figure 2: Oracle Application Server 10*g* Forms Servicves OEM Session page

The "User Sessions" page (Figure 2) provides information about the connected user clients and the actual memory consumption an individual process has on the server. This page was also available in Oracle9iAS Forms Services but in Oracle Application Server 10*g* Forms Services is enhanced to show the application configuration section and the trace group used. Clicking onto the "Configuration Section" link allows the administrator to check the current formsweb.cfg file definitions for this process.

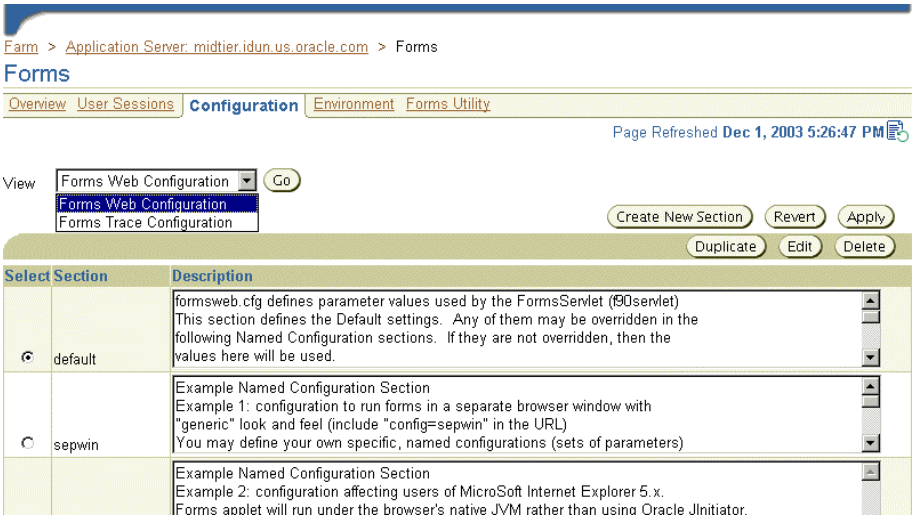

Figure 3: Oracle Application Server 10*g* Forms Servicves OEM Configuration page

The "Configuration" tab (Figure 3) manages the formsweb.cfg application configuration file and the ftrace.cfg trace configuration file that are located in the <OracleAS Home>/ forms90/ server directory.

Changes made by the administrator to this formsweb.cfg file are effective immediately and picked up by any new Forms application request. Existing application instances keep on running unchanged.

The Environment tab page is used to manage the Forms environment variables stored within the default.env file in the forms90/ server directory.

Java 1.4 support on the server and the client

Please visit otn.oracle.com for a list of supported client platforms

Java 2 Enterprise Edition 1.4 (J2EE 1.4) provides new features compared to its predecessor releases. Oracle Application Server 10*g* Forms Services supports J2EE 1.4 enabling you to leverage the latest Java technology stack when building custom Pluggable Java Components (PJC) or server side java classes that are accessed through the Forms Java Importer.

WebUtil (PatchSet)

Moving Forms applications from client-server to the Web means that all application logic now is located on the middle tier server and no longer on the client PC. This however also means that embedded client integration functionality like reading from local text files or MS Office communication now also executes on the middle tier server instead on the users client PC.

WebUtil helps application developers to bring back client integration functionality to the user's Web Browser without recoding the Forms application. WebUtil also covers functionality that in client-server was provided by the "*d2kwutil*" library on Windows.

Last but not least, new functionality, like the browser package, which allows developers to manipulate the Forms external frame and the embedded Forms client Applet, is contained in the WebUtil package to improve the Web client integration.

The aim of WebUtil is to be a utility that any Forms developer can use to carry out complex tasks on the client browser machine by simply coding in PL/ SQL. Although WebUtil itself uses Java extensively there is no need for the programmer to have any understanding of this aspect of the utility. Everything the programmer needs to do is exposed through a standard PL/ SQL library webutil.pll.

All of the components that are needed to support WebUtil are likewise exposed through a simple object group.

Some features of WebUtil, such as client side OLE integration require code to be installed onto each client machine, one of the key features of WebUtil is it's ability to "self-install" in these cases. There is no need for an administrator to configure each and every machine, WebUtil will detect that client-side pieces are missing and will automatically install them.

To obtain a copy of WebUtil, visit OTN at otn.oracle.com/products/forms/htdocs/webutil/webutil.htm

When shipping the base release of Oracle Application Server 10*g* Forms Services, WebUtil is available as a beta version on the Oracle Technology Network (OTN).

Since its beta release in mid of 2003, WebUtil was downloaded several thousand times from OTN and customer feedback and problem reports were logged helping Oracle to improve and fine tune the WebUtil production version. Oracle Forms product development is looking for Forms 10g Patch1 to productize WebUtil, making it an officially and supported part of Oracle Forms.

NEW FEATURES IN FORMS DEVELOPER

Two new features have been added to the Forms building environment Forms Developer.

Support for GIF and JPG icons in the builder

Until Oracle9*i* Forms the icons used on the buttons in Forms were required to be in the '.ico' format on Windows to be shown in the Forms layout editor. This restriction has been lifted in Oracle Forms 10*g* to allow the use of '.jpg' and '.gif ' as well as .ico formats.

As with prior versions of Forms, the icon path needs to be defined in the UI_ICON variable in the Windows registry or the environment setting on UNIX.

A new environment/ registry variable 'UI_ICON_EXTENSION' has been added In Forms 10g to determine the format of the icon expected by the Forms Developer.

- gif
- jpg
- ico

This new features allows application developer to try the icons used on the Web directly when building the Forms application.

Hide Connection Information from the URL when testing Forms

One of the new features in Oracle9*i* Forms was the ability to test a Forms application on the Web using an integrated Oracle Container for J2EE (OC4J) instance that runs Forms Services. The Forms Developer, when clicking the run button or pressing ctrl+r, composes the Forms Services request URL and automatically opens a Browser Window to run the current Forms module on the Web.

The new feature in Oracle Forms 10g is that the 'userid' parameter no longer gets added to the request URL, but instead is hidden within the Forms Servlet session. This adds security in that database username and passwords no longer are exposed in the Browser's URL history.

SUMMARY

Oracle Application Server 10*g* Forms Services and Oracle Forms Developer contain many new features as Oracle continues to enhance and evolve the product. In this release, many of our new features have focused on leveraging the range of

services and facilities of the platform (such as single sign-on and Oracle Enterprise Manager 10g Application Server Control).

Oracle's strategy is to continue integrating Oracle Forms Services with services provided by Oracle Application Server, to support customers that want to run their existing Forms applications seamlessly integrated with applications built with J2EE technologies.

Oracle Forms 10g – New Features
March 2004
Author: Frank Nimphius
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
www.oracle.com

Oracle Corporation provides the software
that powers the internet.