

Oracle Forms Services & Oracle Forms Developer 12c Technical Overview

ORACLE WHITE PAPER | JANUARY 2016





Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.



ORACLE®



Table of Contents

Disclaimer	1
Introduction	1
Rapid Application Development and Deployment	2
Oracle Forms Developer	2
Oracle Fusion Middleware Forms Services	2
Complete Application Framework	3
Build Rich Applications with Oracle Forms Developer	3
RAD Development with Oracle Forms Developer	3
Data Block Wizard	3
Layout Wizard	3
Property Palette	3
Integrated PL/SQL Editor	3
Object Libraries	4
Built-in Packages	4
Support for Unicode	4
Build rich, extensible user interfaces	4
Run Directly to a Browser from the Builder	5
N-Tier “Remote” Debugging	5
The Forms API and Java	6
XML Representation	6
Accessibility	6
Oracle Fusion Middleware Forms Services – Architecture	6

Forms Client	6
Forms Runtime Process	7
Forms Listener Servlet	7
Connection Process Overview	7
Connection Process in Detail	8
Oracle Forms Services benefits	9
Broad Range of Firewalls and Proxies Supported	9
No Protocol Restriction (HTTP/1.1 or HTTP/1.0)	9
No Specific Certificate to Purchase for SSL Deployment	9
Standard Load Balancing Support	9
Support for Java 7 or Java 8 for the Forms Java Client	9
Support for Java 8 on the Middle Tier	10
Global Deployment	10
Run On Any Network	10
Log in Once	10
Run Your Applications in any Timezone	10
Browser Language Detection	11
Tight Integration with the Oracle Database	11
Oracle Database Internationalization Support	11
Performance and Scalability	12
Load Balancing	12
Conclusion	13



Introduction

This paper is designed to provide you with a high-level overview of the features and benefits available through Oracle Forms Developer, and an overview of some of the key points of the Oracle Fusion Middleware Forms Services architecture, a component of the Oracle Fusion Middleware Application Server, and the processes involved when deploying Oracle Forms applications.



Rapid Application Development and Deployment

The rapid growth of e-Commerce and the Internet is accelerating the expansion of user communities and driving demand for higher levels of application performance, reliability, and availability. Business environments are more dynamic than ever before, requiring that applications adapt to changing business requirements in shorter amounts of time. More than ever before, businesses need tools that enable rapid response to changing requirements and rapid and seam-less application deployment.

Oracle Forms Developer

Oracle Forms Developer meets these needs. Oracle Forms Developer is in a class by itself, raising the development bar for applications deployed to such industries as Banking and Finance, Stocks and Bonds, Aerospace, Communications, Manufacturing, Retail, Health, Legal, Government, and Education. With Oracle Forms Developer, business application developers quickly build, in a declarative RAD environment, comprehensive Java client applications that are optimized for the Internet without writing any Java code, and that meet (and exceed) the requirements of professional user communities. These applications are rich, Web-deployed applications available on demand for rapid processing of large amounts of data and rapid completion of complex calculations, analysis, and transactions.

Oracle Forms Developer's integrated set of tools allows business developers to easily and quickly construct sophisticated database forms, and business logic with minimal effort. The development environment provides powerful declarative features, such as wizards, built-ins, and drag-and-drop, for the creation of fully functional applications from database definitions with minimal coding in record time. Oracle Forms Developer also provides an open, extensible user interface model that allows full customization and extension of applications with Java.

Oracle Fusion Middleware Forms Services

Oracle Fusion Middleware Forms Services is an application server and associated services that are optimized for deploying Oracle Forms applications on the Web. Oracle Fusion Middleware Forms Services delivers out-of-the-box functionality and native services to ensure that Oracle Forms applications automatically scale and perform over any network. Oracle Fusion Middleware Forms Services enables rich, extensible Java clients that are optimized for the Web.

Oracle Forms built-in services include transaction management, record caching, record locking, and exception handling. Oracle Fusion Middleware Forms Services built-in services provide a critical infrastructure that developers would otherwise have to code and re-code by hand many times throughout all parts of your application.



Complete Application Framework

Oracle Forms Developer and Oracle Forms Services provide a complete application framework for optimal deployment of Oracle Forms applications on the Internet as well as corporate networks. Together they deliver a Rapid Application Development (RAD) environment and application infrastructure to ensure that your Internet applications scale and perform over any network. The framework is open and extensible and continually evolving, allowing you to:

- » Seamlessly move your applications forward with each new technological wave
- » Easily extend your user interface through native Java with Pluggable Java Components
- » Leverage technologies such as Java and XML through code-based integration

With Oracle Forms Developer and Oracle Fusion Middleware Forms Services, the application framework infrastructure is provided for you, yet you still have the flexibility to leverage the latest technologies within your applications. This allows you to focus on the real value and spend your time thinking about the application business logic and functionality rather than worrying about the application infrastructure.

Build Rich Applications with Oracle Forms Developer

The Oracle Forms client is written and deployed using Java. Oracle Forms Developer enables business developers to build Java applications that are optimized for the Internet without writing any Java code.

RAD Development with Oracle Forms Developer

Both novice and expert developers benefit from Oracle Forms Developers declarative RAD environment. The tool set includes many wizards and utilities to speed application development:

Data Block Wizard

Use the Data Block Wizard to easily create or modify data blocks for use in your application. The Data Block Wizard can be reentered after initial creation of the data block, enabling you to modify an existing data block, even if it was not originally created with the wizard.

Layout Wizard

Use the Layout Wizard to quickly lay out the items of a data block. The wizard displays the items in a frame on a canvas and lays them out in one of several layout styles, which you can manually alter to your own specifications. You can reenter the Layout Wizard after the initial creation of a frame, enabling you to modify an existing frame, even if it was not created with the Layout Wizard.

Property Palette

The Property Palette enables you to set the properties of objects you create in form and menu modules. When you select an object in an editor or in the Object Navigator, the Property Palette updates to show the properties of that object. You can invoke additional Property Palettes as needed, to compare the properties of different objects.

Integrated PL/SQL Editor

The integrated PL/SQL Editor enables you to write PL/SQL code from within the Forms Builder. It provides an interface for editing and debugging applications.

Object Libraries

Object Libraries provide an easy method for reusing objects, and enforcing programming and look-and-feel standards across the entire development organization.

Built-in Packages

Oracle Forms Developer provides several built-in packages that contain many PL/SQL constructs you can reference while building applications or debugging your application code.

Support for Unicode

Oracle Forms Developer and Oracle Fusion Middleware Forms Services provide the means to deliver applications to your users in their native language. Support for Unicode global character set allows multinational corporations to develop a single multilingual application and deploy it to their users worldwide.

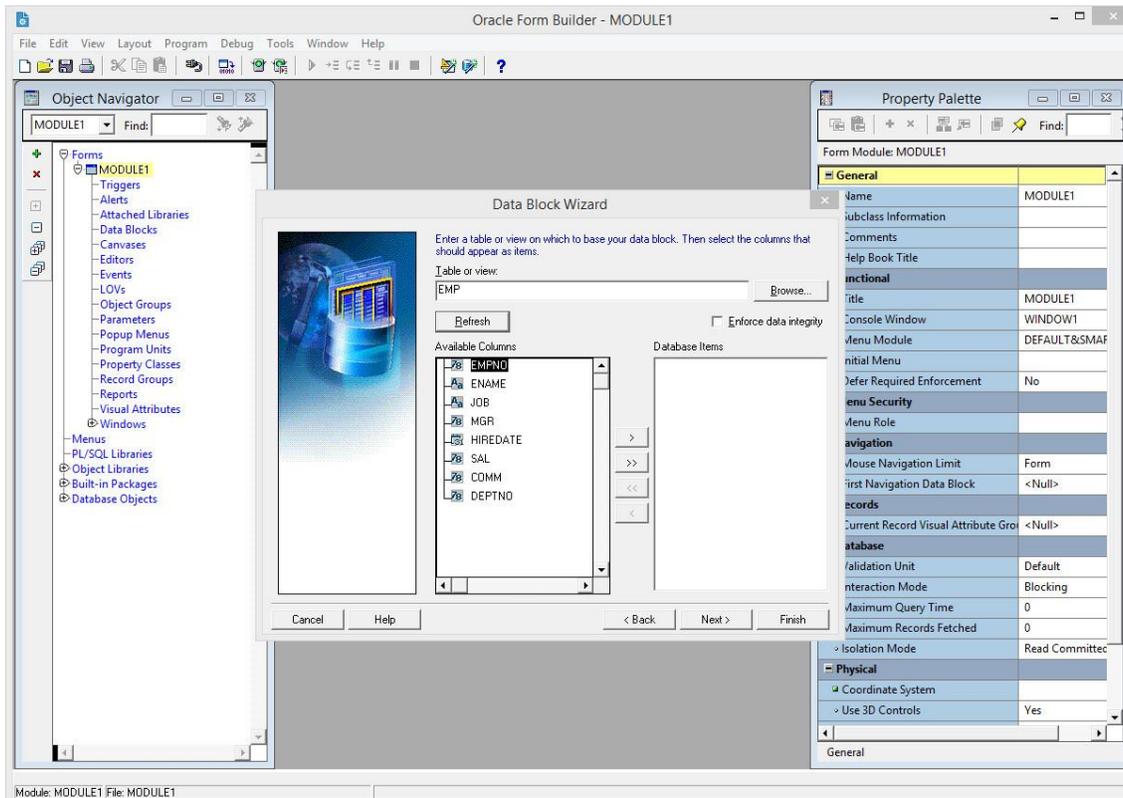


Figure 1: The Data Block Wizard in the Forms Builder.

Build rich, extensible user interfaces

Oracle Forms Developer provides many native widgets and default controls for rapid development of compelling Java UIs with minimal coding. Native widgets and defaults include hierarchical tree controls, tab-pages, check boxes, pop-up lists, tool tips, summary totals, and calculated fields.

Oracle Forms Developer provides an extensible UI through Pluggable Java Components (PJC). You can easily extend and customize your Java user interface beyond the default functionality by integrating your own custom-built Java components into your Forms applications. You'll find a number of sample JavaBeans and PJC's available for download from the Oracle Forms Download area on the Oracle Technology Network (<http://www.oracle.com/technetwork/developer-tools/forms>). These include Web rollover buttons, hyperlinks, and client-side upload functionality. Build Java extensions with Oracle JDeveloper or your favorite 3GL Java development environment, and seamlessly integrate them into your Oracle Forms application.

This is an excellent example of the integration between Oracle JDeveloper and Oracle Forms Developer, and highlights the benefits of the integrated tools offering from Oracle. You have the benefit of the Oracle Forms Developer RAD environment and the scalability of the Oracle Fusion Middleware Forms Services to ensure performance yet can still gain the benefits of open integration with Java to easily customize and extend your applications.

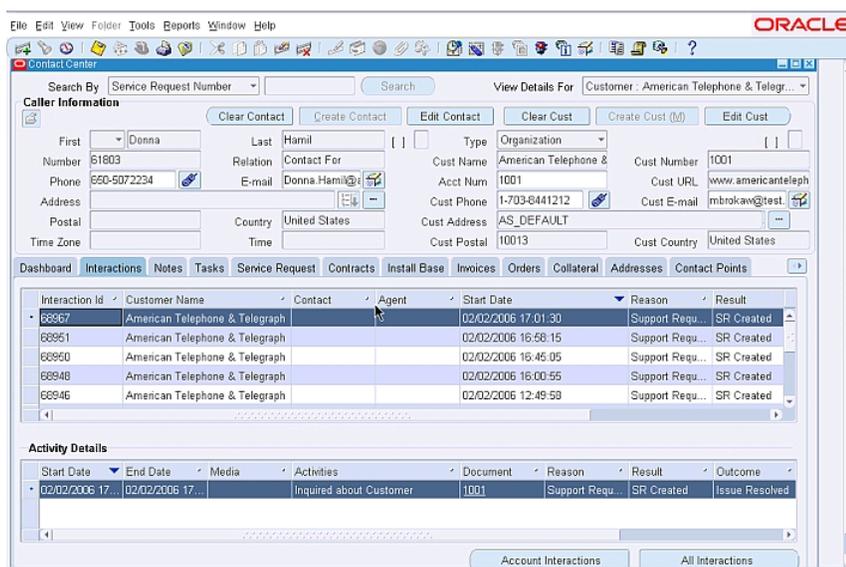


Figure 2: An example of a rich UI that can be accomplished in Oracle Forms

Run Directly to a Browser from the Builder

Within Oracle Forms, when an application is launched from within the Form Builder, it is invoked within your favorite browser or the one you configure within the Builder Preferences. This gives a true reflection of how it will look once deployed, with components such as PJC's fully functional.

N-Tier "Remote" Debugging

You can either run a form directly in debug mode from the Builder, or more impressively, attach dynamically to a remote runtime session over your network. Imagine being able to work cooperatively with an end user, observing the code and stack in their running application as they perform an operation that hits a development bug.



The Forms API and Java

The Forms application-programming interface (API) was introduced with Forms 5.0. This provides a supported way for developers to query and manipulate their Forms modules in batch, without having to open the module in the Forms builder. Because the API is C-based, it is perceived as a “Power User” tool. In Oracle Forms, Oracle is keeping the C API, but also introduced a Java version, the JDAPI (Java Development API). As well as being a language that everyone wants to learn, Java actually makes the API simpler to understand and more compact to use.

XML Representation

The Forms API is a useful tool for carrying out tasks such as dependency analysis, diff-ing, and bulk changes. There are third-party tools on the market using the API to provide exactly these services. But, in an attempt to make such operations even simpler and more accessible in Oracle Forms, Oracle provides a Forms to XML converter. This utility will convert Forms modules into a documented XML format, which can then be edited or transformed into custom output and reports using style sheets.

The XML utility is a two-way operation. You can recreate a Forms module from the XML representation, making that an ideal alternative storage format to the FMT file format that is also provided.

Accessibility

With Oracle Forms, the builder and runtime are compatible with the Job Access With Speech (JAWS) screen reader. Of course, other accessibility requirements are addressed, such as usability without a mouse and compatibility with high contrast color schemes.

Oracle Fusion Middleware Forms Services – Architecture

The Oracle Fusion Middleware Forms Services consists of three components: a Forms Client that is downloaded automatically to the end user’s browser and cached, the Forms Listener Servlet, and the Forms Runtime - on the middle tier.

Forms Client

When a user runs a Forms session, the Forms Client - a thin 100 percent Java applet - dynamically downloads from the Oracle Fusion Middleware Application Server. This generic Java Applet provides the user interface for the associated Forms Runtime process on the middle tier, and handles user interaction and visual feedback such as that generated by navigating between items or checking a checkbox. The same Java applet is used for all Forms application, therefore it is downloaded only once and cached on the client and so is available for subsequent Forms applications.

In order to run a Java applet in a browser, it is necessary to have the Java Plug-in installed. The Plug-in is installed on the client and is platform dependent. Oracle recommends and certifies Java Plug-in 8U51 or newer for this release of Oracle Forms Services.

Oracle Forms now also supports running without a browser, using Java Web Start or the Forms Standalone Launcher. Although Java is still required to be installed on the client machine, installing the Plug-in is optional. Alternatively, an installation of the Java Development Kit (JDK) could be used.

Forms Runtime Process

The Forms Runtime process is the process that maintains a connection to the database on behalf of the Forms Client. The process is created when a user accesses a page containing a Forms application. The process is automatically stopped as soon as the user closes the Forms application or terminates the browser window.

Forms Listener Servlet

The Forms Listener Servlet manages:

- » The creation of a Forms Runtime process for each client when a user requests to run a Forms application.
- » The Forms Listener Servlet is also in charge of stopping the Runtime process as the user closes the Forms application or terminates the browser window.
- » Network communications between the client and its associated Forms Runtime process.

Connection Process Overview

In the Oracle Fusion Middleware Forms Services architecture there is only one connection between the client and the HTTP Listener, much like any web-based application. The HTTP Listener routes the request to the Forms Listener Servlet, which is in charge of routing the requests from the Forms Client to the Forms Runtime.

The communication between the Forms Client and the Forms Runtime always goes through the HTTP Listener leaving the application with only one port open to the Internet.

In this scenario, the client sends HTTP requests and receives HTTP responses from the HTTP Listener process. Because the HTTP Listener acts as the network endpoint for the client, the other server machines and ports are not exposed at the firewall, as shown in the following figure.

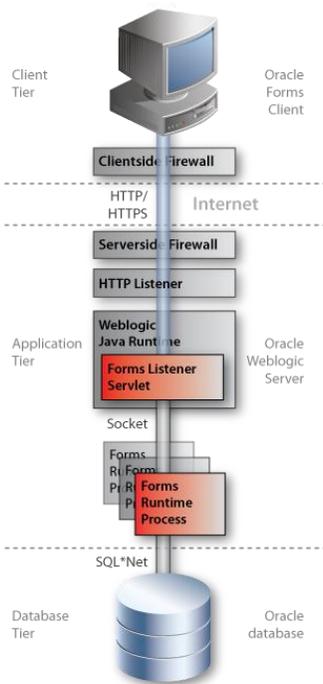


Figure 3: Overview of the architecture

Connection Process in Detail

1. The user chooses a hyperlink from a web page or types a URL directly in the browser.
2. The HTTP Listener interprets the URL received then an HTML page containing an <OBJECT> tag (or JNLP tags, depending on configuration) that describes the Forms Java Client is sent to the client browser. The received URL calls the Forms Servlet to create the HTML page, dynamically based on the html template files located on the application server.
3. The Client receives the HTML file served by the HTTP Listener. The tags in the HTML file will supply the information required to locate the Java Class files that make up the Forms Java Client. Within the tags in the HTML file you would also supply information about the form that should run, and any other parameters that you want to pass to your Forms session, such as the login information. The tag definition also contains instructions on what Forms Services to run and many parameters which can help you to customize aspects of the Java Client such as the look-and-feel, color schemes etc. The HTML file might also contain other HTML attributes such as those to tell the browser to run this particular applet using a particular version of the JRE on the client.
4. The html page sent back to the client also includes the information necessary to launch the JRE (or Web Start – depending on the configuration). Once the client side JVM has started, it uses several of the parameters included in the page to request the needed archive (.jar) files and determine the startup Java class file within them. This archive will have an extension of .JAR and can be best thought of as a .ZIP file containing all of individual CLASS files required by the Applet. The use of a JAR file speeds up the download of the Java Client and enables caching on the client for subsequent calls. The ARCHIVE parameter defines which .JAR file(s) should be used. The client JVM (a component of the Java Plugin and Java Web Start engine) will carry out the additional step of comparing the timestamp of the cached Forms Client Java code (if a cached version exists) with the server version. A new jar download will only occur if the server is newer than the timestamped jar stored in the plug-in cache.
5. Once the JAR files are downloaded (if not already present) to the Java applet starts.
6. The Java Client applet sends a request to start a Forms session through the HTTP Listener to the Forms Listener Servlet. The Forms Listener Servlet is defined by the serverURL parameter in the HTML file.
7. After receiving the connection request from the Java Client, the Forms Listener Servlet starts a new Forms Runtime process for this client. You can define user specific environments for each runtime process by setting the Servlet initialization envFile parameter in the configuration file – formsweb.cfg - to a specific environment file. In the figure we have created three files (French.env, Swedish.env and English.env) to support different languages; the default file is default.env and this file contains for example the Forms ORACLE_HOME and FORMS_PATH.
8. The Forms Runtime process allocated to this client, loads the module specified in the HTML file and any libraries and menus that are required by that form. All communication between the Forms Client and the Forms Runtime process is passed through the Forms Listener Servlet.
9. The user is prompted for database login information, if this had not already been supplied, and the connection to the database server is established.
10. The user is now ready to work.

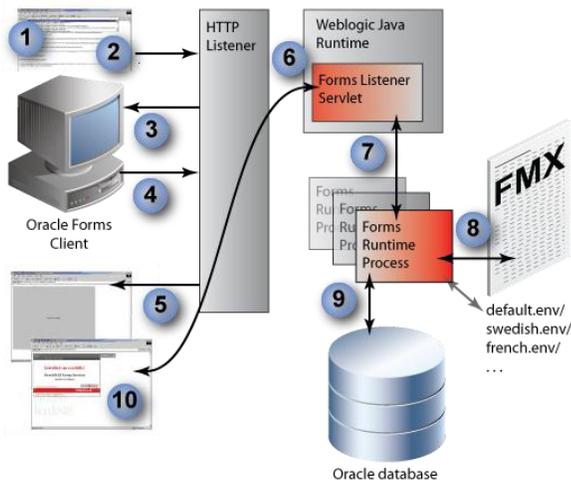


Figure 4: Overview of the connection process.

Oracle Forms Services benefits

The Forms Listener Servlet is designed to allow a robust and standard deployment of Forms applications on the Internet. The Forms Listener Servlet provides the following benefits:

Broad Range of Firewalls and Proxies Supported

Because the client browser always communicates with the HTTP Listener using HTTP or HTTPS (there is no direct connection between the client and the Forms Runtime process), this architecture supports any firewall or proxy that can work with a standard servlet using servlet sessions.

No Protocol Restriction (HTTP/1.1 or HTTP/1.0)

Since standard communication is used, both HTTP/1.0 and HTTP/1.1 are supported. HTTP/1.1 will give better performance, but at the expense of consuming more resources on the server.

No Specific Certificate to Purchase for SSL Deployment

In the case of deployment using SSL (secure sockets layer), the HTTPS connection occurs between the client browser and HTTP Listener. Therefore, there are no specific security configuration requirements at the Forms Server level.

Standard Load Balancing Support

This architecture allows you to use standard load balancing techniques, such as hardware based load balancing, reverse proxy, and standard Servlet load balancing. (More information is available later in this document.)

Support for Java 7 or Java 8 for the Forms Java Client

With the 12c (12.2.1) release, the Forms Java Client is able to use Java 7 or Java 8 (refer to the Product Certification Matrix for a list of exact versions). Those who are writing Pluggable Java Components or trying to integrate the Forms Java Client with other Java applets or JavaBeans will be able to use the full power of Java 7 or 8 APIs.



Support for Java 8 on the Middle Tier

The Java Importer feature is a key part of the Forms integration strategy, providing simple-to-use Java integration from PL/SQL. Again, the Java Importer can be used with Java 8, making it possible to integrate Forms applications with the latest external programs and services, and call out to Java classes on the middle tier.

Global Deployment

More and more, companies are deploying applications to wider audiences and across a variety of networks and systems. The complexity in network and server configurations can be daunting. Oracle Forms, a web-only release, helps you face the challenges of deploying your applications globally.

Run On Any Network

The Forms Listener Servlet architecture allows you to deploy your Oracle Forms applications in a robust and standard manner. With the Java servlet, you can run your applications on any network: Internet, intranet, or extranet. Only standard ports in your firewall need to be opened and authenticating proxies are supported.

For secure connections, no extra SSL Certificate is required, as Forms Services will use the HTTP Listener's certificate. In addition, the Forms Listener Servlet architecture removes the administration overhead of managing the Forms Listener as well as allowing you to take ad-vantage of the entire standard load balancing capabilities provided by your HTTP Listener or hardware.

Log in Once

Increasingly, companies are moving their information infrastructure online. Often, each user must maintain a separate user name and password for each application they access. Maintaining multiple accounts and passwords for each user is expensive, insecure, and most of all, impractical.

Oracle Forms can use the authentication services (SSO) provided by Oracle Access Manager and Web Gate, available as components of Oracle Fusion Middleware, which integrated with various LDAP servers (refer to the OAM documentation for a list of supported LDAP servers). Using Single Sign-On eliminates the need for users to remember multiple passwords because they only have to log in once. Moreover, no additional coding is required in your Forms application modules.

Run Your Applications in any Timezone

An important, but often overlooked, consideration when deploying applications is the simple concept of time. When a user records a time value in an application, what exactly do they mean? The time where they are? The time where Forms Services is running or the time on the database server?

Imagine the scenario where a call center employee based in England is on the phone to a customer reporting the time of a credit card theft whilst on holiday in Hawaii, with the data stored in a database in New York. Then if that needs to be matched up with a fraudulent purchase made over the Internet to a website in Singapore, you can see how a universal concept of time is important.

Oracle Forms supports the ability to define the time zones at all levels of your application so that the correct time is stored in the database.

Browser Language Detection

Oracle Forms is providing a feature that automatically detects a client's browser language setting. You can use this feature to allow users to access an application through the same URL, and automatically redirect them to the translation that matches their language or the language set in their browser.

Tight Integration with the Oracle Database

With over 60 percent of the global market share, Oracle is the worldwide leader in database servers across all industries. Oracle offers superior manageability, high availability, and ease of use. It provides integrated data management for all Internet content within your organization. Oracle Forms Developer is specifically designed and optimized to build Oracle transactional database applications. Oracle Forms Developer is designed for the Oracle database. It delivers the following services for you natively—services you would otherwise have to code by hand:

- » Connects to and maintains a connection to the Oracle database.
- » Queries and handles a large number of records on demand.
- » Locks database records on demand.
- » Generates code that automatically supports multi-user locking scenarios.
- » Manages inserts, updates, and deletes automatically.
- » Allows programmatic manipulation of sets of records for a developer.
- » Communicates transactions efficiently to the database in an atomic fashion, meaning that a transaction is either fully committed or not at all.
- » Automatically handles communication with database Advanced Queuing queues.
- » Automatically handles logins when database proxy users are used for deployment.

Use the Data Block Wizard to automatically link your application to tables in your database and easily build a complex Master-Detail Web form with operations automatically synchronized between two or more data sets. Use the form to query, insert, update, and delete data and to immediately preview your changes on the Web. With this seamless integration between Oracle Forms Developer and Oracle database you can build Internet applications that use Resource Management, Advanced Queuing, Subscription, Distributed Queries, Partitioning, and Parallel Server to more effectively share resources between applications and to improve application performance and scalability. Improve the performance of queries across multiple homogenous and heterogeneous environments through optimized distributed query plans and other replication enhancements. Implement fine-grained, centralized access control and auditing through Oracle security policy management features.

Oracle Database Internationalization Support

Forms supports a feature of the Oracle database that ensures the length of character variables used in your multi-language applications will accommodate the language in which the application is deployed. For example, consider the following declaration:

```
my_string VARCHAR2(100);
```

This variable holds 100 bytes. In a standard English or Western European character set such as US7ASCII or WE8ISO8869P1, this variable would hold 100 characters. However, when using a multi-byte character set, such as JA16SJIS or UTF8, this variable may hold 50 characters or less. The new character semantics feature resolves this problem by allowing you to declare variables by the absolute number of characters required, irrespective of the underlying character set.



Performance and Scalability

A lot of work has been done with the Oracle Fusion Middleware Forms Services to make it as performant as possible from end to end and giving you performance that you would expect from a Client-Server implementation, but with the vastly reduced administrative overhead of Web deployment.

Load Balancing

Since the architecture in Oracle Forms is standard based servlet architecture there is no need of a specific Forms load balancing utility. Oracle Fusion Middleware Forms Services utilizes the WebLogic Server infrastructure and Oracle HTTP Server, which both offer load balancing features.

This architecture also allows you to use standard load balancing techniques, such as hardware based load balancing, managed server clustering, and reverse proxying. For more detailed information on how to setup load balancing please refer to the Oracle Fusion Middleware documentation.



Conclusion

The demands placed on corporate information systems to readily adapt to changing business requirements and emerging technologies is overwhelming. Development teams are continually faced with demands for enhanced application functionality, improved user interfaces, and more complex, high-performance deployment configurations. Oracle Forms Developer and Oracle Fusion Middleware Forms Services provide a scalable, flexible architecture to automatically deliver high-performance, enterprise class applications to all of your users. Oracle customers are using Oracle Fusion Middleware Forms Services.

Services to support thousands of users around the world, and benchmarks have proven exceptional performance by running thousands of concurrent users. By leveraging the Oracle Internet Platform through its openness and inherent scalability, Oracle Forms Developer and Oracle Fusion Middleware Forms Services provides an integrated delivery environment for Internet applications that automatically scale and perform.



Oracle Corporation, World Headquarters

500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries

Phone: +1.650.506.7000
Fax: +1.650.506.7200

CONNECT WITH US

-  blogs.oracle.com/oracle
-  facebook.com/oracle
-  twitter.com/oracle
-  oracle.com

Integrated Cloud Applications & Platform Services

Copyright © 2016, Oracle and/or its affiliates. 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 and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0615

Oracle Forms Services & Oracle Forms Developer 12c: Technical Overview
January 2016
Author: Michael Ferrante
Contributing Authors: Oracle Product Management

 Oracle is committed to developing practices and products that help protect the environment