ORACLE FUSION MIDDLEWARE

Oracle Application Server Forms 10*g* Release 2 (10.1.2.0.2) Technical Overview

An Oracle White Paper
August 2005

ORACLE

ORACLE FUSION MIDDLEWARE

Oracle Application Server Forms 10*g* Release 2 (10.1.2.0.2) Technical Overview

ORACLE FUSION MIDDLEWARE

Oracle Application Server Forms 10*g* Release 2 (10.1.2.0.2) Technical Overview

ABOUT THIS PAPER

This paper is designed to provide you with an high-level overview of the features and benefits available through Oracle Forms Developer, a component of Oracle Developer Suite, and an overview of some of the key points of the Oracle Application Server Forms Services architecture, a component of the Oracle Application Server (OracleAS) 10*g* Release 2, and the processes involved when deploying Oracle Forms applications over the Web.

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 seamless 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. Key customers and partners in these industries include Century Consultants, CMiC, Eurostar, Metro, HP Italy, Annamas, Retek, LIMITrader, and Keystone Solutions. 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 Java client 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 builders 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.

"Among the benefits: better response times to client inquiries and more billable hours on the clock, thanks to faster, more accurate time reporting. Time is now recorded daily on screen rather than weekly on paper. With fee earners entering data directly into the system, many potential errors, including faulty transcribing of matter numbers and descriptions and misplaced timesheets, are eliminated."

DJ Freeman uses Keystone Practice Management Solutions, http://www.keystone-solutions.com.

"As part of Oracle's Developer Suite, Oracle Forms proved to be an excellent reliable and scalable solution for deploying our application over the Internet. The thin client concept and central administration approach cut costs for us and our customers."

- Bob Magan, Development Manager, Century Consultants, http://www.centuryltd.com/

ORACLE APPLICATION SERVER FORMS SERVICES

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

OracleAS Forms Services built-in services include transaction management, record caching, record locking, and exception handling. 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 OracleAS Forms Services provide a complete application framework for optimal deployment of Oracle Forms applications on the Internet. 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 OracleAS 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 JAVA APPLICATIONS WITH ORACLE FORMS DEVELOPER

RAD Development with Oracle Forms Developer

Both novice and expert developers benefit from Oracle Forms Developers declarative RAD environment. Oracle Forms Developer enables business developers to build Java applications that are optimized for the Internet without writing any Java code. 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 a graphical interface for editing and debugging applications.

Object Libraries

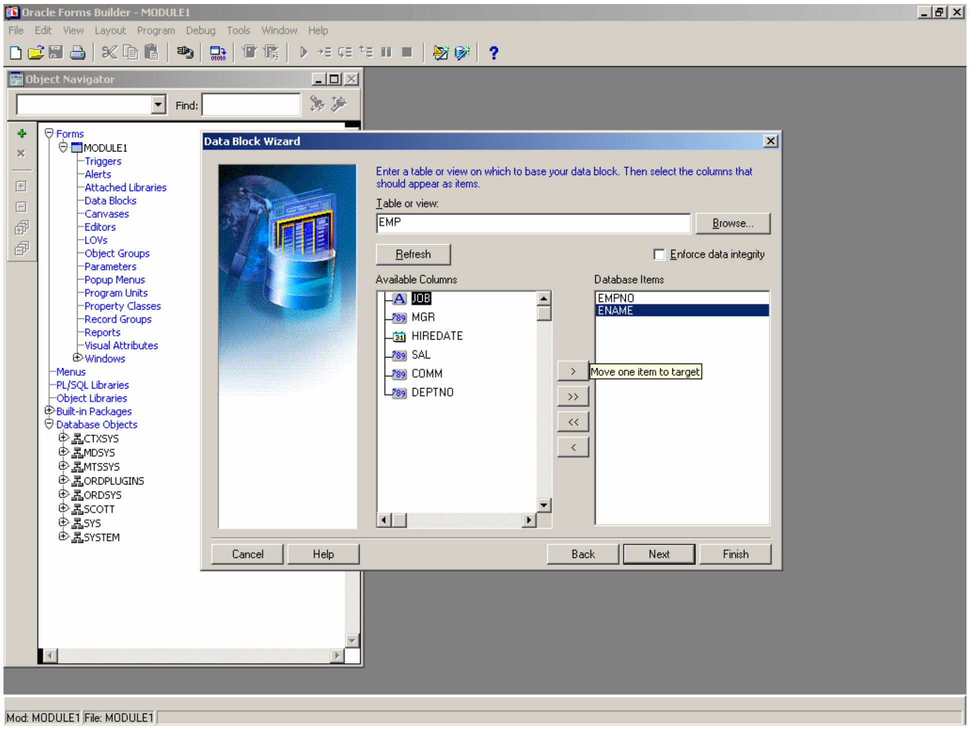
Object Libraries provide an easy method for reusing objects, and enforcing programming and look-and-feel standards across the entire development organization.

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 OracleAS 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.



Enter a table or view on which to base your data block. Then select the columns that should appear as items.

Table or view: EMP

□ Enforce data integrity

Available Columns: JOB, MGR, HIREDATE, SAL, COMM, DEPTNO

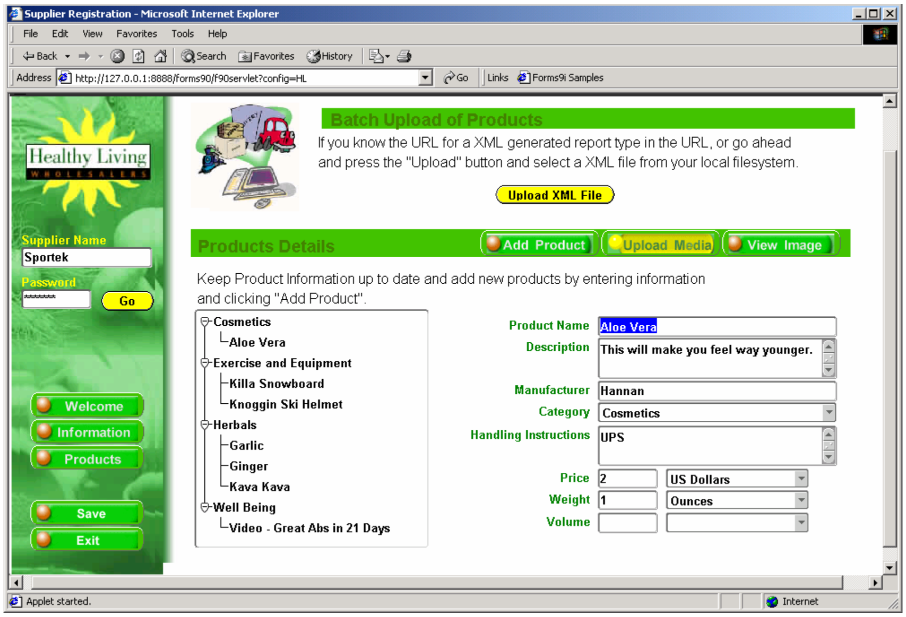
Database Items: EMPNO, ENAME

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 (PJCs). 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 PJCs available for download from the Oracle Technology Network (http://otn.oracle.com/products/forms/content.html). 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. The Oracle JDeveloper PJC Wizard helps you wrapper your custom Java classes to extend your Forms applications.

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 OracleAS Forms Services to ensure performance yet can still gain the benefits of open integration with Java to easily customize and extend your applications.



Address: http://127.0.0.1:8888/forms90/f90servlet?config=HL

Batch Upload of Products

If you know the URL for a XML generated report type in the URL, or go ahead and press the "Upload" button and select a XML file from your local filesystem.

Supplier Name: Sportek

Password: *******

Products Details

Keep Product Information up to date and add new products by entering information and clicking "Add Product".

- Cosmetics
  - Aloe Vera
- Exercise and Equipment
  - Killa Snowboard
  - Knoggin Ski Helmet
- Herbals
  - Garlic
  - Ginger
  - Kava Kava
- Well Being
  - Video - Great Abs in 21 Days

Product Name: Aloe Vera

Description: This will make you feel way younger.

Manufacturer: Hannan

Category: Cosmetics

Handling Instructions: UPS

Price: 2 US Dollars

Weight: 1 Ounces

Volume:

Figure above - The Healthy Living application, a demo built with Oracle Forms Developer, showcases a rich Java UI that uses native features of Forms Developer and leverages Java to extend the UI through Pluggable Java Components.

Run Directly in a Browser

Within Oracle Forms, when the application is run from within the builder, it is invoked within your favorite browser. This gives a true reflection of how it will look once deployed, with components such as PJCs 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 intranet. 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 goes 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 Forms Oracle is keeping the C API, but also introducing a new 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. As a bonus, the Oracle Developer Suite product suite comes with a Java Development Environment, in the form of Oracle JDeveloper, so you already have everything you need to build utilities with the Java API.

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 is also providing a Forms to XML converter. This utility converts 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 at 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 APPLICATION SERVER FORMS SERVICES – ARCHITECTURE

The OracleAS 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 (Java Applet)

When a user runs a Forms session, the Forms Client - a thin 100 percent Java Applet - dynamically downloads from the Oracle 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 any 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 a Java Virtual Machine (JVM) installed. The JVM is installed on the client and is platform dependent. On Windows platforms Oracle provides JInitiator, which has benefits over other JVMs when running a Forms application on the web.

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:

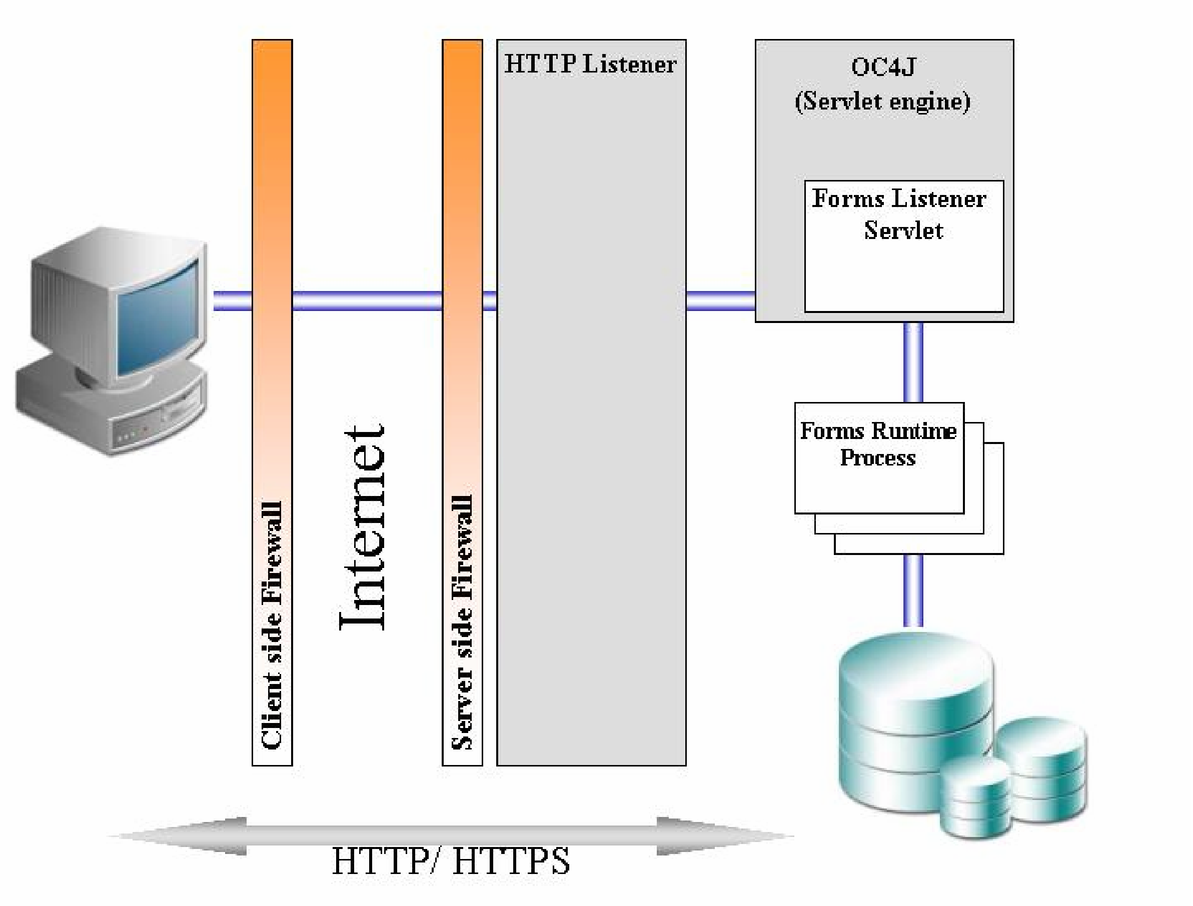
- The creation of a Forms Runtime process for each client when a user requests to run a Forms application.
- The Forms Listener Servlet is also in charge of stopping the Runtime process as the user closes the Forms application or terminates the Browser window.
- Network communications between the client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

Connection process overview

In the OracleAS 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 go through the HTTP Listener leaving us with only one port open to 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.



Connection process in detail

See also figure below

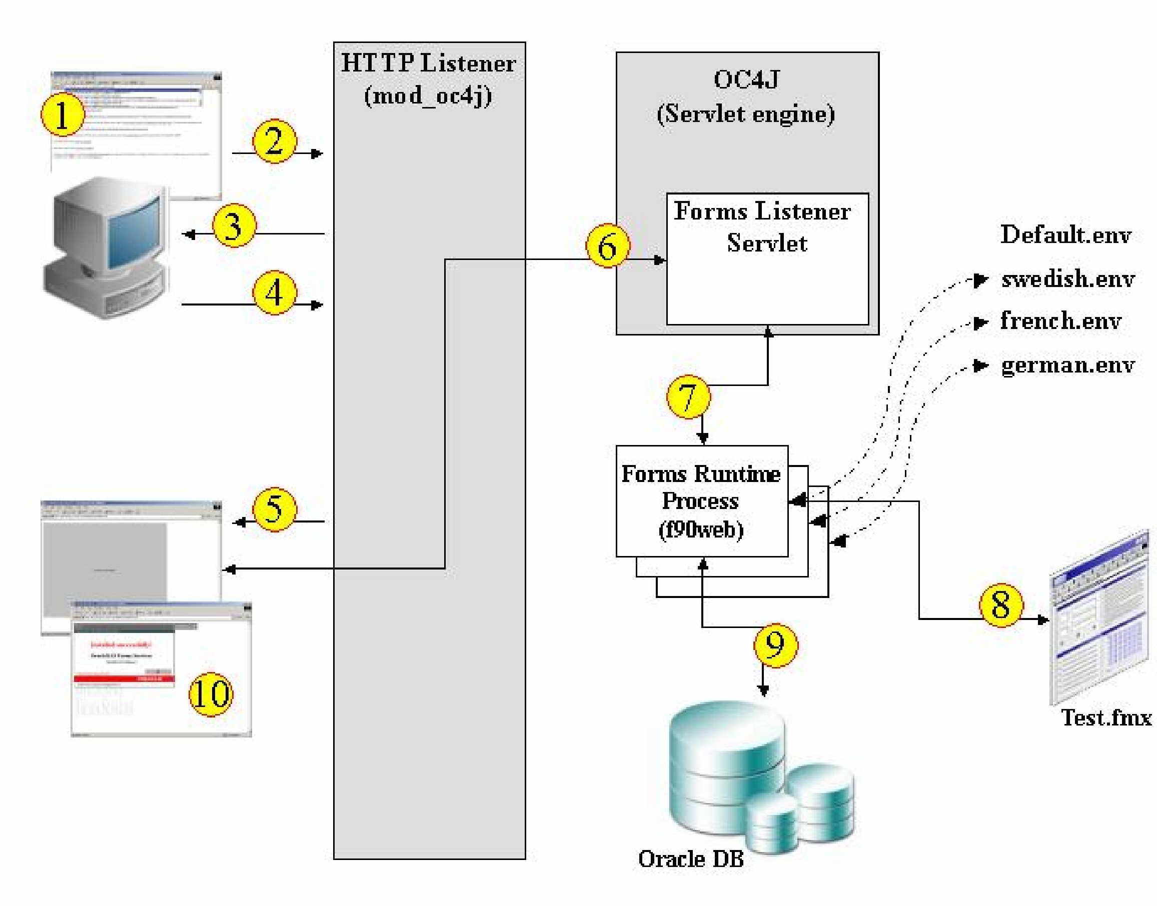
1. The user chooses a Link from a web page or types a URL directly in the Browser
2. The HTTP Listener interprets the URL that is passed and displays an HTML page containing an <APPLET> tag that describes the Forms Java Client to the Browser. The URL that is passed calls the Forms Servlet to create an HTML page dynamically based on the base.html files located on the web server.
3. The Client receives the HTML file served by the HTTP Listener. The <APPLET> tag in the HTML file supplies the information required to locate the Java Class files that make up the Forms Java Client. Within the APPLET tag 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 APPLET 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 tags such as those to tell the browser to run this particular applet using the JInitiator Plug-in (see JInitiator).
4. The Browser then asks the HTTP Listener for the Java Class files from the location specified in the HTML file. The CODEBASE parameter in the HTML file is used to define this. The files may be downloaded individually or as an "Archive". This archive has an extension of .JAR or .CAB 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 or CAB file speeds up the download of the Java Client and enables caching on the client for subsequent calls. The ARCHIVE parameter defines which (if any) .JAR or CAB file should be used. If this particular HTML file specified that JInitiator should be used, then JInitiator will carry out the additional step of checking the version of the Forms Client Java code available on the HTTP Listener and will only download it if it turns out to be newer that any version that JInitiator currently has cached.
5. The CLASS or JAR files are downloaded (if not already present) to the Browser and 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's APPLET tag.
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.



ORACLE APPLICATION SERVER 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/manage 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 1.4.x JDK for the Forms Java Client

With the OracleAS Forms 10*g* release we are upgrading the Forms Java Client to be able to use the 1.4.x version of the Java Runtime Environment (JRE). In most cases this will make little difference to customers, but 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 the 1.4.x Java APIs for those clients that use the 1.4.x JVM.

Support for 1.4.x JRE in 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 the 1.4.x version of the Java Runtime, 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 this new 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 advantage of the entire standard load balancing capabilities provided by your HTTP Listener or hardware.

Log in Only 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 provided by the Login Server and mod_osso included with OracleAS, which are integrated with the Oracle Internet Directory (OID) and is LDAP-compatible. 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.

Deploy to Multiple Languages

To assist you in translating your Forms applications to other languages, Oracle TranslationHub keeps a repository of your translations and lets you preview the Forms while they are being translated. Using Forms Services, you can deploy a Forms application in many different languages with a single installation.

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.

Forms and Reports Services Installation

For customers who wish to run only Forms and/or Reports Services, a separate installation configuration is available. The Forms and Reports Services CD comprises only OracleAS J2EE core components + Forms Services + Reports Services. There is no Infrastructure to install and configure, just the minimum needed to deploy your Forms and Reports applications.

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. The Oracle RDBMS 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 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

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 the 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 new 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 OracleAS Forms Services to make it as preferment 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 overheads of Web Deployment, and no client software installs!

Details of how to leverage the maximum scalability and performance from the OracleAS Forms Services can be found in the White Paper: Oracle Forms Capacity Planning Guide downloadable from the Oracle Technology Network (http://otn.oracle.com/products/forms/content.html).

LOAD BALANCING

Since the architecture in Oracle Forms is standard based servlet architecture there is no need of a specific Forms Load balancing utility. OracleAS Forms Services utilizes the entire infrastructure of Oracle Application Server, which includes the Apache/OC4J load balancing and WebCache load balancing.

This architecture also allows you to use standard load balancing techniques, such as hardware based load balancing, and reverse proxy.

For more detailed information on how to setup Oracle Application Server and Load balancing please refer to the Oracle Application Server documentation.

ORACLE JINITIATOR

Oracle JInitiator is Oracle's version of Sun's Java Plug-In, which provides the ability to specify the use of a specific Java Virtual Machine on the client instead of using the browser's default JVM. Oracle JInitiator runs as a plug-in for Netscape Navigator and as an ActiveX component for Internet Explorer; allowing customers to run OracleAS Forms Services applications using Netscape Navigator or Internet Explorer.

The Oracle JInitiator provides these major benefits:

- Allows the latest Oracle-certified JVM to run in older browser releases.
- Ensures a consistent JVM between different browsers.
- Provides functional extensions to the basic JVM such as HTTPS/SSL support.
- Provides a reliable deployment platform. Oracle JInitiator has been thoroughly tested and certified for use with the OracleAS Forms Services and for Oracle Applications E-Business suite.
- JInitiator is the preferred deployment environment. Application class files are automatically cached by JInitiator, providing faster application start-up.
- JInitiator is a self-installing and self-updating deployment environment. Oracle JInitiator automatically installs and updates itself like a plug-in or an Active-X component. Local cached application class files will be automatically updated from the application server based on a date-time stamp comparison.

PROTECT YOUR INVESTMENT

Whenever a new software technology is introduced, early adopters often sacrifice the benefits of mature, productive, and powerful development tools in the rush to exploit and deploy the latest thing. Does this mean that cutting-edge businesses must choose between the latest technology and the best tools?

No!

By choosing the right development tools from the right tool vendor, businesses can significantly reduce the technology learning curve, dramatically increase productivity, and actually shorten the time to market while producing higher quality, more reliable, more scalable applications that effectively exploit the benefits inherent in new technology.

By *the right tool*s, we mean, among others, Oracle Forms.

Throughout its history, Oracle has responded to technological advances with tools that enable customers to preserve their investment in business logic and applications development while exploiting the benefits of each new technological wave. Oracle Forms Developer has enabled companies to leverage character-based technology into client/server, and client/server technology into Web-deployed forms. OracleAS Forms Services provides the underlying platform and built-in services that enable immediate exploitation of the benefits of each new technology while allowing redeployment of legacy forms with just a few changes to server configuration.

Oracle Forms Developer and OracleAS Forms Services provide all of the traditional productivity benefits you'd expect in a Rapid Application Development (RAD) environment. Additionally, they have consistently provided insulation against technology shifts by allowing you to migrate your applications forward to take advantage of each new technology as it emerges. Protect your investment and migrate to the Internet with Oracle Forms Developer and OracleAS Forms Services.

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 OracleAS Forms Services provide a scalable, flexible architecture to automatically deliver high-performance, enterprise-class applications to all of your users. Oracle customers are using OracleAS Forms 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 OracleAS Forms Services provides an integrated delivery environment for Internet applications that automatically scale and perform.

ORACLE FUSION MIDDLEWARE

Oracle Application Server Forms 10*g* Release 2 (10.1.2.0.2) Technical Overview
August 2005
Author: Robin Zimmermann
Contributing Authors: Regis Louis, Duncan Mills, Grant Ronald, Frank Nimphius

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