RAPID APPLICATION DEVELOPMENT (RAD) AND DEPLOYMENT USING ORACLE™

Krassimira Shvertner

Abstract: This paper is designed to provide with a high-level overview of the features and benefits available through Oracle9i Forms Developer, a component of Oracle9i Developer Suite (Oracle9iDS), and an overview of some of the key points of the Oracle9iAS Forms Services architecture, a component of the Oracle9i Application Server (Oracle9iAS), and the processes involved when deploying Oracle9i Forms applications over the Web. Oracle9i Reports Developer is also considered.

Key words: Oracle Servers, Web Technologies, Forms Developer, Reports Developer.

INTRODUCTION

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.

ORACLE9i FORMS DEVELOPER

Oracle9i Forms Developer is the successor of the Oracle Forms family of RAD tools. Oracle9i 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 in Bulgaria the Bulgarian Telecommunication Company, BNB etc. With Oracle9i 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.

Oracle9i Forms Developer’s integrated set of builders allows business developers easily and quickly to 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. Oracle9i Forms Developer also provides an open, extensible user interface model that allows full customization and extension of applications with Java.

Both novice and expert developers benefit from Oracle9i Forms Developers declarative RAD environment. Oracle9i 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 the application. The Data Block Wizard can be reentered after initial creation of the data block, enabling 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 can be manually altered using own specifications. The developer can reenter the Layout Wizard after the initial creation of a frame, enabling...
the developer to modify an existing frame, even if it was not created with the Layout Wizard.

- **Property Palette** The Property Palette enables the developer to set the properties of objects created in form and menu modules. When the developer selects an object in an editor or in the Object Navigator, the Property Palette updates to show the properties of that object. Additional Property Palettes can be invoked as needed, to compare the properties of different objects.

- **Integrated PL/SQL Editor** The integrated PL/SQL Editor enables the developer 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** Oracle9i Forms Developer provides several built-in packages that contain many PL/SQL constructs the developer can reference while building applications or debugging the application code.

- **Support for Unicode** Oracle9i Forms Developer and Oracle9i/AS Forms Services provide the means to deliver applications to the 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.

- **Build Rich Java Applications** Oracle9i 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. Oracle9i Forms Developer provides an extensible UI through Pluggable Java Components (PJC). The developer can easily extend and customize the Java user interface beyond the default functionality by integrating developers own custom-built Java components into the Forms applications.

- **Run Directly in a Browser** Within Oracle9i Forms, when the application is run from within the builder, it is invoked within developer’s favorite browser. This gives a true reflection of how it will look once deployed, with components such as PJC fully functional.

**ORACLE9i AS FORMS SERVICES**

Oracle9i/AS Forms Services is an application server and associated services that are optimized for deploying Oracle9i Forms applications on the Web. Oracle9i/AS Forms Services delivers out-of-the-box functionality and native services to ensure that Oracle9i Forms applications automatically scale and perform over any network. Oracle9i/AS Forms Services enables rich, extensible Java clients that are optimized for the Web. Oracle9i/AS Forms Services built-in services include transaction management, record caching, record locking, and exception handling. Oracle9i/AS 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 the application.

Oracle9i Forms Developer and Oracle9i/AS Forms Services provide a complete application framework for optimal deployment of Oracle9i Forms applications on the Internet. Together they deliver a Rapid Application Development (RAD) environment and application infrastructure to ensure that the Internet applications scale and perform over any network. The framework is open and extensible and continually evolving, allowing the developer to:

- Seamlessly move his applications forward with each new technological wave ·
Easily extend his user interface through native Java with Pluggable Java Components.

Leverage technologies such as Java and XML through code-based integration. With Oracle9i Forms Developer and Oracle9iAS Forms Services, the application framework infrastructure is provided, yet the developer still have the flexibility to leverage the latest technologies within the applications. This allows the developer to focus on the real value and spend more time thinking about the application business logic and functionality rather than worrying about the application infrastructure.

**ORACLE9iAS FORMS SERVICES – ARCHITECTURE**

The Oracle9iAS 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 Oracle9i 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 Listener Servlet**

The Forms Listener Servlet manages:

- communications between the client and its associated Forms Runtime process

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

Forms Runtime process

Listener Servlet starts a new Forms Runtime process for this client. The developer 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 (e.g. 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 FORMS90_PATH. 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.

The business logic itself is saved in Forms .fmx files and PL/SQL library files. The Forms Web Runtime executes both file types and connects to the database using Net9. While it executes the business logic, the Web runtime sends metadata messages to the client instructing it to render the user interface.

Because all Forms application modules are executed on the middle tier server, Forms services automatically shared static application resources, thus reducing the memory requirements for each individual user process.

ORACLE9i/REPORTS

Oracle9i Reports is Oracle’s award-winning, high-fidelity enterprise reporting tool. It enables businesses to give immediate access to information to all levels within and outside of the organization in an unrivaled scalable and secure environment.

Oracle9i Reports consists of Oracle9i Reports Developer (a component of the Oracle9i Developer Suite) and Oracle9iAS Reports Services (a component of the Oracle9i Application Server). Oracle9i Reports Developer uses a declarative, document-centric development model to help the J2EE developer rapidly create high-fidelity Web and paper reports against any data source (including Oracle9i, JDBC, XML, text files, and Oracle9i OLAP). While allowing developers to leverage the latest J2EE technologies such as JSP and XML, Oracle9iAS Reports Services securely publishes these reports in any format (including HTML, XML, PDF, delimited text, Postscript, PCL, and RTF) to any destination (including e-mail, Web browser, Oracle9iAS).

Figure 2. Reports creation workflow
Building a Report using the Data Wizard to add data to a sample Web page

When the developer builds a report, he must first build a data model by selecting the data that he wants to use in the report. The steps in this section show how to use the Data Wizard to build a data model for a Web report. The data model the developer creates makes the data from the sample schema available to be used in the report.

To add data to an existing HTML page:
1. Open the Data Wizard to define a layout and add a data model. In the Object Navigator, double-click the Data Model node. In the Data Model view, right-click the canvas, then choose Data Wizard from the pop-up menu.
2. If the Welcome page displays, click Next.
3. On the Query name page, click Next to accept the default name.
4. On the Data Source page, make sure SQL Query is selected, then click Next.
5. On the Data page, the developer can paste a SQL query into the SQL Statement box.

Oracle9iAS Reports Services

Oracle9i Application Server is a comprehensive and integrated application server that runs any Web site, portal, or Internet application. It enables the developers to make applications available from Web browsers, mobile devices, and command lines. Oracle9iAS Reports Services is the Reports publishing component of Oracle9iAS. It is an enterprise reporting service for producing high quality production reports that dynamically retrieve, format, and distribute any data, in any format, anywhere. The developer can use Oracle9iAS Reports Services to publish in both Web-based and non-Web-based environments. Oracle9iAS Reports Services provides a scalable, flexible architecture for the distribution and automated management of report generation engines on the same server and across multiple servers. Additionally, it caches report outputs for reuse on similar requests. It enables the developer to run reports on both local and remote application servers and to implement a multi-tiered architecture for running the reports.

When used in a Web environment, the Oracle9iAS Reports Services architecture consists of four tiers:
- The client tier (a Web browser)
- The Web server tier
- The Oracle9iAS Reports Services tier
- The data tier, including databases and all other data sources

When used in a non-Web environment, there are three tiers (a Web server being unnecessary):
- The client tier (a small, proprietary application on each client machine)
- Oracle9iAS Reports Services tier
- The data tier, including databases and pluggable data sources

Oracle9iAS Reports Services handles client requests to run reports by entering all requests into a job queue. When one of the server’s engines becomes available, the next job in the queue is dispatched to run. As the number of jobs in the queue increases, the server can start more engines until it reaches the maximum limit specified in the server configuration. Similarly, engines are shut down automatically after having been idle for a period of time that the administrator specifies.

Oracle9iAS Reports Services keeps track of all jobs submitted to the server, including jobs that are running, scheduled to run, finished, or failed. The Reports Queue Manager (Windows), the Reports Queue Viewer (UNIX), the showjobs command (Web), and the Reports Services pages in Oracle Enterprise Manager (OEM) enable to view information on when jobs are scheduled, queued, started, finished, and failed, as well as the job output and the final status of the report.
APPLICATION WORK

Using Oracle Forms 9i, Oracle Reports 9i together with OAS9i Forms and Reports servers is developed integrated software hardware fault diagnosis system for quality of services support in telecommunication network. It is intended to fine faults, to localize and identify the faults, to monitor and control the faults and supervise the faults reparation. The system works in real exploitation as Center 130 in all regions of BTC.

Program realization is module designing, flexible and parameter ability. The system is prepared to different configurations and application using parameters. The system is prepared to different test units, digital exchanges and structures in BTC.

The functional structure of fault diagnostic system in telecommunication network includes the follow:
- fault requests registrations;
- distance automatic tests;
- input the test results;
- fault diagnosis;
- monitoring and control of the faults;
- reparation control;
- group test of the equipment;
- group test using lists;
- forecast of future faults in equipment;
- visualization of shames, graphics, failure equipment and sections;
- access to elements data, topology and composition of network, data for exchanges capacity and subscribers data (name, address).

To support of main system functions there are developed the follow modules:
- Fault requests registration module, distance automatic test and visualization of subscriber data;
- Module for monitoring, supervision and control of the faults;
- Test module and evaluation the results;
- Module for test requests management;
- Module for fault diagnosis;
- Module for group faults registration;
- Module for monitoring and control of the group faults;
- Module for closing the group faults;
- Module for requests of group tests in time using lists or criterion;
- Management schedule control module;
- Module for input system data.

The fault diagnostic system is a structural program and support different type of work places. The system users have their own password and user role to work with the system. Depending on the role, the user can see different type of data on his screen.

The system make statistical reports for faults and scheduling reports, reports for employers, reports for elements with many faults, prophylactic recommendation for elements and equipment in telecommunication network.

CONCLUSIONS

Oracle Forms 9i, Oracle Reports 9i together with OAS9i Forms and Reports servers establish a powerful RAD environment. They use instructional approach for designing of Forms and Reports on the web and client/server environment. This approach especially by Forms server led to minimized number of packets exchanged between the client and the middle tier in contrast to the classical web design approach. The productivity of forms and reports design is very high.
REFERENCES
[1] Oracle 9i Application Server Release 2, Forms Services Overview, Oracle Corporation
[2] Oracle9iAS Forms Services and Oracle9i Forms Developer, Technical Overview, Oracle Corporation

ABOUT THE AUTHOR
Scient. Researcher Krassimira Shvertner, PhD, Center for Information Technologies in Telecommunication, Bulgarian Telecommunication Company, Phone: +359 2 8705 076, -mail: Shvertner.cits@btc.bg.