

Oracle9i Forms and J2EE: A Business Case for Application Development

*An Oracle Whitepaper
July 2003*

Oracle9i Forms and J2EE: A Business Case for Application Development

| | |
|--|---|
| Introduction | 3 |
| Introduction to the Business Case | 3 |
| Background..... | 3 |
| Plans for expansion | 4 |
| Consider the Choices | 4 |
| Customer advisors accessing the new stock data | 5 |
| Building new applications using Oracle Forms | 5 |
| Warehouse staff updating data..... | 5 |
| Building new applications using J2EE | 5 |
| Customers checking order status on the Web..... | 6 |
| Web services..... | 6 |
| Building a J2EE application that calls Web services..... | 6 |
| Sharing business logic between Forms and Java | 7 |
| A resounding success | 8 |
| Conclusion..... | 8 |

Oracle9i Forms and J2EE: A Business Case for Application Development

INTRODUCTION

As customers build and enhance their enterprise applications, they will continue to seek out the best tools and technologies available to meet their customer and business requirements. At the same time, IT managers have to maintain a competitive edge. Oracle Forms has long been a tool of choice for building Internet-based database applications. However, Java 2 Enterprise Edition (J2EE) has presented a new choice to enterprise application developers and one that appears to manifest itself as a crossroads for Forms developers.

So, as Oracle Forms professionals, are we really at a crossroads, where we can only take one path and leave behind all of our investment in Forms, or can we have the best of both worlds?

This paper discusses the options that are available at these crossroads, provides technical pointers, and looks at a business case that concludes that now is the time to see J2EE as a complement to existing Oracle Forms applications rather than competing with them.

INTRODUCTION TO THE BUSINESS CASE

Mr. Appsavvy is the IT manager of the fictional “Summit” company and has been following the trends of software development for many years. He has been involved in building Oracle Forms applications for the past six years and his last major project involved moving the Summit client/server Oracle Forms application to the Web. This migration was completed successfully six months ago and now the company is considering its next phase of development. However, he is hearing more and more about J2EE, and since many of his new developers have Java experience, he now wonders if this is the direction in which they should be moving.

Background

Summit has a long history of using Oracle Forms. Its principal Oracle Forms application is used to manage customers and their orders. Its workflow starts when customers call the Summit office to place orders. The service representative takes

the order using the Oracle Forms application, enters details about the customer, the order, then submits the order for fulfillment (delivery, billing, etc.).

Traditionally, Summit has used Oracle Forms because of its application requirements: the application is database intensive, operators require a rich user interface with instant validation and complex screen manipulation, and developers have skills that have evolved from their history with the Oracle database and are comfortable with PL/SQL.

Plans for expansion

Business has been good for Summit and they are ready to expand their operations and execute new development on their IT infrastructure. Summit's immediate plans are to have a new inventory control application where warehouse employees can access it using hand held devices. Specific inventory data will be accessible to customer services representatives as well so that they can verify availability. This application can also allow customers the ability to track their orders on the Web. The order will be dispatched by their courier partner, "Speedy Packages", which means that Summit customer service representatives, as well as their customers, want to be able to integrate with the courier firm to check on the last stages of order fulfillment.

CONSIDER THE CHOICES

Mr. Appsavvy knows from experience that the success of this development project is based on meeting the requirements of the business and not the underlying technologies or tools. He knows that he must consider issues such as:

- What are the skills of his development team?
- What style of application do his users require?
 - Functionally rich and complex user front end?
 - "Self Service" style of application?
- Does Summit have a corporate strategy that pushes them towards a specific technology?
- Will the decision impact licensing costs?
- Where is the company's current technological investment?
- Does the application require interoperability?
- What are the migration issues for rolling out this new application?

Given that Mr. Appsavvy's decision will be based on these kinds of issues, he decides not to get preoccupied by the technologies but instead, focuses on the business requirements.

Customer advisors accessing the new stock data

The new inventory application needs to be accessible by the customer service representatives in a highly interactive way when they are taking customer orders. Initial prototyping has shown that a tree control that allows the user to drill down into the multiple categories of products and then view the details of each stock item is the most efficient way of presenting the data to these users. It has also been noted that the inventory data must be accessible as part of the order transaction. This new application must also work with the current customer order (Oracle Forms) application.

Building new applications using Oracle Forms

Mr. Appsavvy identifies the main requirements for this area of development as:

- Rich and sophisticated user interaction
- Tight integration with the existing Oracle Forms application
- Utilize existing database schemas

Given that the screen flow between the current Forms application and the new inventory data is so tightly coupled, he decides it should really form part of the same application. Feedback from users when upgrading the current application to the Web shows that the Forms style of UI meets the requirements of his end users. Further, he still has a strong pool of Oracle Forms developers who performed the upgrade to the Web.

Consequently, Mr. Appsavvy has decided that he will continue to use Oracle Forms to build this *part* of the application.

Warehouse staff updating data

His next consideration is how the inventory data is maintained in the warehouse itself. Warehouse employees will be using hand held devices to maintain information about inventory levels, which need to be accessed via a lightweight wireless device. Mr. Appsavvy summarizes the requirements as:

- Application accessed via hand held devices
- Utilize existing database schemas

Building new applications using J2EE

Mr. Appsavvy knows that for this part of the business, the hardware devices that will be used by the end users will drive his decision. In this case, the profile of a hand held application does not match that of an Oracle Forms application and so he now must investigate the options available in the Java world.

While there are many tools around in the market which can be used to build this style of application, Mr. Appsavvy primarily investigates Oracle9i JDeveloper because it comes as part of the Oracle9i Developer Suite (Oracle9iDS), of which Forms is a component.

Mr. Appsavvy has recognized that while a technology meets the needs of your business and users, it makes sense to continue to exploit that success. Oracle Forms has a proven track record in "Summit" and he sees no reason to change while Oracle Forms continues to meet their requirements.

While being an award winning J2EE tool in its own right, Mr. Appsavvy has recognized that while his company is currently licensing Oracle9iDS, it makes sense to consider Oracle9i JDeveloper. However, his evaluation of the tool has shown it to be ideal for development of J2EE applications.

He quickly realizes that JDeveloper provides frameworks to easily access the existing Summit database as well as providing tools for building wireless front ends

He also has the opportunity to exploit the skills of his newly recruited Java developers.

Mr. Appsavvy is also using the database as an integration point by sharing data and tables (and possibly application logic) in the database.

My Appsavvy thus decides to use Oracle9i JDeveloper to build the warehouse application which will access the same data that is used by the customer services representatives in the inventory control element of their Forms application.

Customers checking order status on the Web

Summit wants its customers to be able to check the status of their orders on the Web. Summit also would like to build an application where customers can log in and view details about their orders. In addition, “Speedy Packages”, who Summit uses as its courier partner, have a package tracking system that is written in .Net.

Therefore, this application will have to check to see if the order has been dispatched to the courier, and if so, will call out to the Web service that “Speedy Packages” provides for tracking orders in its system.

Again, Mr Appsavvy considers the business requirements:

- “Self Service” style of application
- Integration with .Net via Web services

Given the integration with Web services, Mr. Websavvy decides to investigate more about how Web services can be used in business.

Web services

Web services are causing much excitement in the application development world because of the way business functionality can be packaged as components that can be accessed on the Web. The popular “Stock Quote” and “Currency Conversion” examples are often used as demos of Web services. These demos only give a bite size description of the possibilities that are really almost boundless. For more details on Web services, visit the Web Services Center on OTN¹.

Building a J2EE application that calls Web services

Mr. Appsavvy sees that exploiting the Web service that is published by “Speedy Packages” gives him the perfect “hook” into another businesses application. Now he must decide how the rest of the application is built.

First of all, he considers whether his development team should write this new application using Oracle Forms given the success they have had when deploying other Oracle Forms applications on the Web. Although a Web-deployed Oracle

¹ <http://otn.oracle.com/tech/webservices/content.html>

M.r Appsavvy sees that Web services are a natural integration point. He has identified that the Web service provided by the courier can be exploited without having to understand the platform or language used to implement it.

Forms application could meet this requirement, the traditional Oracle Forms UI would probably be “overkill” given that the user only wants to view some data and have minimal interaction with the application.

Mr. Appsavvy understands that the self-service style of application fits in with the profile of a JSP/HTML front end interfacing with an application that queries the database to check the status of the order. The application will also provide seamless integration with the courier’s business.

Mr. Appsavvy immediately sees that the combination of a JSP/HTML front end to an application that accesses business logic through a Web service ideally lends itself to the J2EE framework.

Sharing business logic between Forms and Java

However, during user acceptance testing, the customer service representatives indicate that they need to access the order status from their Oracle Forms application as well. Because Mr. Appsavvy has made the decision to build the order status application in a J2EE framework, he can now published this functionality as a Web service and integrate with the Forms application.

Creating a Web service.

JDeveloper allows you to easily publish your business logic as Web services using simple wizards. PL/SQL, Enterprise Java Beans, and any Java class can be published as Web services.

And to make it even easier, JDeveloper provides a wizard for creating a Java client class that can be used for calling a Web service.

Calling a Web service from Forms

Given that the integration point between Oracle Forms and the Web service is through a Java client class, the Oracle Forms Java Importer is used to create a simple PL/SQL wrapper around the Web service client to integrate Forms and Web services. More details on calling a Web service from Forms are available on OTN².

Mr. Appsavvy’s foresight in partitioning the application means that since this information needs to be shared across Summit’s own Forms and J2EE applications, publishing it as a Web service provides the ideal solution.

Also, by doing this he is confirming that he does not have to chose either Forms or J2EE but instead use the best technology to fit his business needs

Thus, business logic is being shared across the technologies to meet a business need in an efficient manner.

² http://otn.oracle.com/products/forms/htdocs/Forms_WebService_How_To.htm

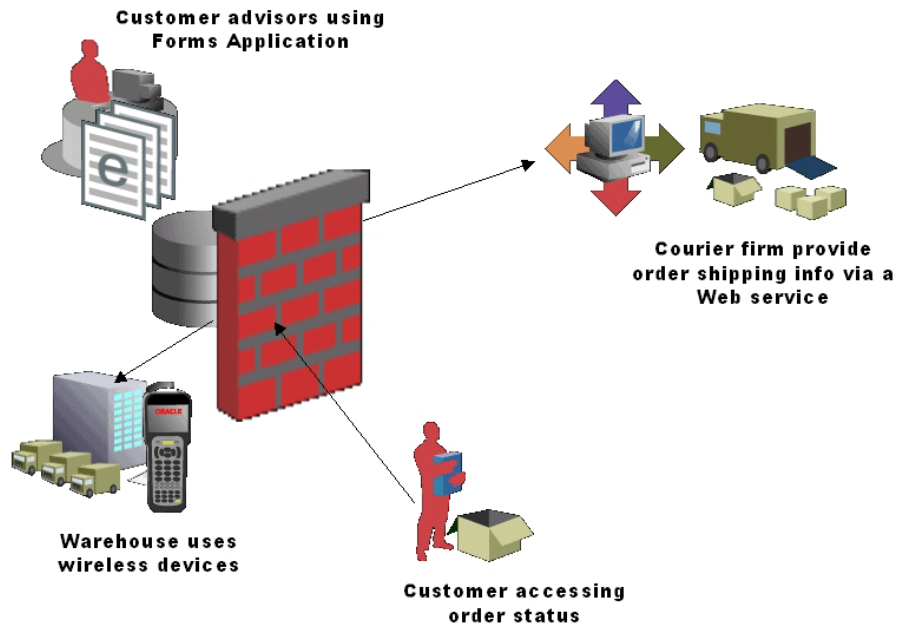


Figure 1- The Summit application using Forms and J2EE

A resounding success

Summit has now implemented their business requirements using a mix of Oracle9i Forms and J2EE. It has preserved and extended its existing investment in their current Oracle Forms application as well as using the powerful J2EE framework to leverage facilities that are not readily available in the Oracle Forms world. Summit has also shared business logic across the technology scopes and integrated with third-party applications. A success all around!!

CONCLUSION

Oracle Forms has a long and illustrious history evolving from character mode, to client/server, to Web-deployed applications. J2EE is an exciting new option for application development that the Oracle Forms world is embracing.

Now is the time to see J2EE as complementing your existing Oracle Forms applications, not competing with them.



Oracle9i Forms and J2EE: A Business Case for Application Development

July 2003

Author: Grant Ronald

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.

Oracle is a registered trademark of Oracle Corporation. Various
product and service names referenced herein may be trademarks
of Oracle Corporation. All other product and service names
mentioned may be trademarks of their respective owners.

Copyright © 2003 Oracle

All rights reserved.