Efficiently Migrating Oracle® Forms to Java™/J2EE

A review of circumstances facing IT management when considering the move to Java EE.

Prepared by Fujitsu Legacy Modernization and OpenText™
Table of Contents

Introduction ........................................................ 3
Benefits of Java .................................................... 4
Moving off of Oracle Forms ....................................... 4
OpenText™ Composer Ciphersoft ................................ 5
Faster time to market and quicker ROI ......................... 7
Solution .................................................................. 8
Fujitsu Services .................................................... 10
Fujitsu Legacy Modernisation ..................................... 10
Introduction

Many organizations have made substantial investments in existing applications. These tried and true applications have provided business value and the required functionality for several years. But what happens when the organization wants to leverage its existing investment in data and business logic with newer applications? Typically nothing. Leaving the organization in a state of complete arrested development. It makes far more business sense to implement new technology where the impact can be the greatest (such as front-end of the application) and the cost impact will be minimal. The goal is to leverage existing application systems and data while providing new functionality. Any technical decision that maximizes the business impact and minimizes the cost and risk of failure is always a good business decision.

As Oracle continues to promote its 10g and 11g environments, and migrates its own applications to Java, many organizations are contemplating using the new web capabilities that are being provided.

Some organizations have business reasons to migrate immediately to a Java environment. This may be driven by the requirement to use JDBC connectivity between applications, or because there is a specific business reason within the organization that requires an application to run in Java. Therefore, organizations are trying to determine whether they should stay with Oracle Forms as a development environment or migrate to Java.

As organizations develop modernization strategies and roadmaps for their Oracle Forms environments, the applications’ sizes and complexities, and retiring Oracle Forms resources, such as developers and lack of accurate documentation, are all factors that must be considered. Many organizations consider migrating to Java because of its lower development costs, integration capabilities, and their requirement for an open, standards-based environment. It is important that organizations understand the business reasons to migrate their applications and determine if the effort and dollars spent will justify the costs of a conversion to Java. Understanding the risks of migration and the benefits of Java is imperative to ensure an organization is not spending precious IT dollars on an alternative that does not make the best business sense.

As organizations review the differences between migrating to Java and developing in 10g, some key components may assist in the decision:
Benefits of Java

Java offers a robust, interactive environment. It is one of the most powerful Object Oriented Programming languages available and one of the most “open” technologies available. Java conforms both to its own standardized (and published) specifications as well as to other industry standards. JDBC database connectivity provides a standardized interface for relational databases, providing a greater level of database independence and portability. It also provides platform independence, allowing an organization to use the most efficient and effective hardware available.

Why is Java superior?

Java is the only technology that provides a fully interactive GUI interface for the web. The Java architecture was designed with security in mind and not as an afterthought. This provides a simplified and consistent means of protecting IT assets. Java also provides features that allow programming to become easier and more powerful.

These include multi-threading capabilities:

- Automatic “garbage collection” (for efficient use of memory)
- Standardized error trapping and detection
- Distributed processing capabilities

Oracle’s strategic direction

Oracle’s acquisition of Sun Microsystems combines two powerhouses focused on delivering complete, open and integrated products to customers worldwide. As Oracle is one of the leading providers of Java-based application infrastructure middleware, its long-term application development strategy remains based on the Java platform.

Oracle’s goals include:

- Pooling server-side Java virtual machines to reduce the memory footprint of applications that call middle-tier Java
- Reduced application pre-starting
- Performance and scalability on the Web
- Expanding the scope and depth of the Forms management tuning and problem diagnosis facilities within Oracle® Enterprise Manager
- Extensible client and middle-tier Java integration (Java Importer and Pluggable Java Component Interface)

As the momentum toward the Java platform grows and the future of Oracle Forms becomes more restricted, the decision to migrate to the Java platform should be commenced prior to facing the situation and cost of extended support.

Moving off of Oracle Forms

The options facing Oracle Forms installation owners can be summarized as follows:

1. Do nothing – stay with Oracle Forms and undertake ‘extended support’ once support for Oracle Forms 11g ceases
2. Rewrite the applications
3. Migrate the Oracle Forms applications to Java EE
Rewrite the application

There is a subtle reason that programmers often want to throw away code and start over. They think the old code is inefficient and that rewriting the application will make it work better—they are often wrong. And while the old code may have problems it’s often harder to read code than to write it. This is why code reuse is often so difficult. Programmers tend to write their own functions because it is easier and more fun than figuring out how the existing functions work, and to make improvements.

Rewriting applications from scratch is the most expensive and time-consuming option available. Many organizations believe that rewriting existing applications will solve their problems because they will work better or because they need additional functionality. The problem becomes maintaining and ensuring the new logic is bug free, users are trained in and understand the new application, and the application actually meets the organizational needs. They also don’t take into account how many applications become abandoned midway through building the application, wasting valuable resources.

This alternative is by far the riskiest. It is not only time-consuming, but also shares a number of other risks that can significantly affect the viability of the project. These include human error, lack of resources or skill set, lack of understanding of the business logic that is required and disruptions to the business operations. In addition, escalating costs usually coincide with code re-write projects. Organizations have started their Oracle Forms to Java conversions manually only to realize that the time difference between manual and automatic conversions is significant.

Migration to Java is beneficial if:

- Java has been selected as the future development environment for the organization
- Lower application development costs is a business imperative (Java development provides much lower development costs)
- Legacy Oracle Forms application requires integration with other applications developed in other environments
- Other applications within the organization use Java technology
- Application Servers or IDE are already in use within the organization, which require independence
- The use of open source technology is beneficial to the organization
- Provision of choices within the client’s environment is required (i.e., HTML, DHTML, and browsers)
- The use of Service Oriented Architecture (SOA) is beneficial to the organization
- The organization has determined that proprietary technology is no longer beneficial

OpenText™ Composer Ciphersoft

OpenText Composer Ciphersoft is an automated tool that converts existing Oracle Forms and PL/SQL applications to open and standards-based technologies, such as Java, XML and browser-based solutions. The product will optionally convert the PL/SQL within the database to EJB Session Beans and Bean managed entity beans to standard Java classes, or leaves it in the database unchanged but invoked by external Java classes. The Oracle Forms themselves are converted to XML-based displays that run within the Java-based interface. The product performs this function with a limited amount of operator intervention.

A conversion project is easily managed and has been shown to save 90 percent of the time and 80 percent of costs in the conversion process. The converted application is 100 percent compliant with J2EE standards. It provides a complete migration of Oracle Forms and PL/SQL to Java and no “wrapping.”
OpenText Composer Ciphersoft provides Java and XML code that is portable, maintainable, reusable and web-centric and uses database connectivity via JDBC. OpenText Composer CipherSoft supports multi-tier architecture and enables the use of all Java specific development tools available today and in the future (i.e. designers, modellers, testing tools, source code archives, change management software, application servers, JSP and JSF integration tools, Java compilers.)

Ease of maintenance is facilitated due to convenient separation of responsibilities among the logical tiers of your application and the code generated is 100 percent Java code. OpenText Composer Ciphersoft also converts PL/SQL packages and libraries to EJB’s or Java classes and can be deployed on any Java EE compliant web server or run as standalone applets.

The migration process is as follows:

The left side of the above illustration is a high level representation of an Oracle Forms application, with a user interface, business logic, Oracle Forms framework and the database. The framework automatically handles things used repeatedly in Oracle Forms applications, such as trigger execution logic, global and system variables, synchronization between fields on the screen and fields in the database and data access. Providing this framework is what gave Oracle Forms developers such a high level of productivity. The implementation details are transparent so the developer can focus on the business logic.

In a migration to Java or any other low level programming language, these basic components are still needed for every Oracle Forms application. Recreating the basic components of the Oracle Forms Framework would require a lot of low level coding just to handle basic operations such as synchronization between fields on the screen and fields in the database, even before focusing on the business logic of the application itself. The benefit of an automated migration solution is that all of this has been handled, providing a 90 percent reduction in cost and time vs. a rewrite of the underlying components.
When deploying to the Java EE environment, the scenario appears as follows:

When an Oracle Forms application is migrated to Java, the application is deployed to an application server environment, such as Oracle 10g, Oracle® Fusion (11g) or Apache® Tomcat. For the Java environment, applications are deployed to an application server, such as Oracle Application Server in 10g or the Fusion Application Server in 11g. The steps required in the deployment process can vary based on the network and security requirements. For applications outside of the intranet, additional parameters for security need to be incorporated (for instance single sign-on and encryption). In Java, the application is compiled into a collection of files, which consist of Java code, classes, etc.

At deployment time, these are “zipped up” into a WAR file, which is deployed to the application server. Organizations with an application server usually have a process in place for deploying these WAR files on their application server. OpenText Composer CipherSoft provides simple step-by-step instructions for deploying on the Oracle Application Server or Tomcat for applications migrated with the software.

A different approach must be used for applications that are modified frequently. Partial deployment cannot be done in Java, as could be done in the Oracle Forms scenario. Instead, organizations use source management software to manage the build-release process. Many of the source management solutions have a utility or mechanism that can automate the process so the application can be pushed out on a daily basis with modified files without a lot of manual intervention.

**Faster Time to Market and Quicker ROI**

The most immediate benefit of using OpenText Composer CipherSoft is time to market. The software provided by OpenText and partnered with Fujitsu, one of the largest and most experienced Systems Integration companies worldwide, so the conversion and migration of applications can be accomplished quickly without tying up developer and technical resources for months. Rapid adoption of the new platform enables organizations to begin recognising the ROI identified in the cost justification model at project inception, and to avoid the extended support costs that the future plans for Oracle Forms will incur.

**Customer overview**

The customer is a large commercial bank (“the Bank”) in the United States with more than 2,800 banking offices and 5,000 ATMs. The Bank provides a comprehensive line of banking, brokerage, insurance, investment, mortgage, trust and payment services products to consumers, businesses and institutions.
Challenge

The Bank purchased a company with several Oracle Forms-based applications and more than 250 Forms. Business requirements dictated moving the applications to the Java platform so they could integrate with other Java applications within the larger organization.

The Bank reviewed the options for converting these Oracle Forms applications to Java. Through this review, its goal was to find an efficient and effective way to move these applications into the Java EE environment with minimal disruption to the application’s users. Additionally, the Bank wanted to save time and money getting the applications to Java, while retaining the legacy investments made in the Oracle Forms environment.

The Bank did not have expertise in the Oracle Forms environment, and its technological capabilities and strategic technology direction did not include Oracle Forms. It faced the challenge to move the applications to Java, minimize the disruption of service for its customers and achieve an aggressive migration timeline.

The Bank evaluated options including, a complete re-write of each application to simply converting the data without the business logic, UI layers of the applications. However, these options were not as attractive due to timelines and the requirement for little to no disruption time for the customer. The costs included with customers learning a new system were not reasonable.

The Bank initiated a methodology for moving the applications to Java, including:

- Identify software direction for the applications
- Work with vendors to explore the various vendor-driven options
- Evaluate the options
- Analyze the options
- Determine a suitable solution

One of the biggest challenges for the Bank was its lack of domain knowledge about the Oracle Forms applications. Therefore, it required a solution that would provide like-for-like conversion of applications, including the business logic, User Interface and data, without involving the subject matter experts.

Solution

The Bank selected the OpenText Composer Ciphersoft conversion tool, and contracted with the OpenText migration team to do the actual conversion work. With its team of Java developers, it needed to get these acquired Oracle Forms applications to a platform where the team could easily maintain and enhance the applications going forward.

OpenText Composer Ciphersoft, which converts Oracle Forms to 100 percent Java code, not only saved the Bank time, but maintained the business logic of the original applications, which was a critical requirement. If the Bank had chosen to rewrite the application, all of the business requirements would need to be recreated and redeveloped – creating a long and complex process. The automated conversion tool provided a similar look and feel and, therefore no new learning curve for the users. Following the conversion, the entire application ran in the Java environment, and the data integrity was maintained in the existing Oracle database. Also the converted applications, once moved to production, were completely maintainable by bank personnel. OpenText Composer Ciphersoft has no proprietary code and no perpetual licensing. Having maintainable code also allowed them the opportunity to extend and optimise the applications, giving their users added functionality.
Why OpenText Composer Ciphersoft was Chosen

OpenText Composer Ciphersoft is the only software tool to convert more than 95 percent of the original Oracle Forms application, including all of the business logic, as well as the user presentation layer and the data access. The tool provides the choice to have the same look and feel of the original application, or to define a new GUI for the application. Any time an automated tool can be used, it reduces risk and testing time, as an automated converter always produces the same results. Therefore, once a common routine is generated and tested, the results of that routine will be consistent throughout the application.

Additionally, the proposal provided a fixed price solution, as well as the staff to complete the project and support knowledge transfer to Bank personnel as they learned the new application. It also included resources for additional customization of the product and resultant migrated application.

Results

By using the conversion technology, the Bank could have all its Oracle Forms applications migrated to Java cleanly and quickly while minimizing risk and dramatically reducing the application’s time to production. The largest benefit seen from the customer was the ability to convert the code using a black-box approach for the conversion.

Coordination and communication were essential between the development team for the Bank and the migration team. Both sides appointed project managers to be central points of contact and they held regular status meetings. Interactive internet sessions were scheduled to show and discuss issues, allowing both teams to see what the other saw. Also, detailed testing issue templates were used and updated for each application. Testers entered details of any issues encountered so that OpenText developers could easily duplicate the issues, resulting in quicker resolutions.

During the course of the project, the Bank ran into several unsupported Oracle functions, and/or uses of built-in functions. In every case, the functionality was evaluated, and based on the occurrences and necessity, was built into the product and the application. Some of the key issues encountered and implemented were Key Mapping and Hot Key usage, new properties added to components to allow developers to tweak performance on pages used for heavy data entry, and configurable parameters for buffering data for Lists of Values and table entries.

The OpenText Composer Ciphersoft converted Java code was not based on the same server-base as the client’s platform. Through the appointed contacts, issues stemming from this discrepancy were resolved in a timely manner.

The Bank performed extensive performance and load balancing testing before moving the converted applications into production. All issues were resolved through the combined efforts of on-site bank personnel and the project team prior to going into production, thus giving the client a high degree of confidence on the “go-live” dates. Once the applications were up in production, feedback was extremely positive – with very smooth transition.

The Bank achieved its project metrics to eliminate the Oracle Forms applications and transition the applications to the Java platform on time and within budget. With the OpenText Composer Ciphersoft product and team, this was achieved. The core business functionality remains the same and users are easily able to use the new Java applications without a learning curve.

According to the Development Manager at the Bank, “Using OpenText Composer Ciphersoft, we were able to accurately convert and efficiently migrate our legacy applications. We were able to meet our timelines and provide an application that is positioned for future growth.”
Fujitsu Services

Fujitsu’s Legacy Modernisation offering is a global service charged with assisting clients to improve the technological roadmap of their legacy IT portfolio.

A legacy system is considered to be any system that is currently in production. These systems will normally have received significant investment over the years and closely reflect the organization’s required functionality. Legacy systems are very often significant assets and represent competitive advantages in an organization’s marketplace.

As systems are made up of a wide array of technology, such as platforms, languages, architectures, database, etc., it is only natural that some of these technologies will be superseded or not be considered core to an organization’s technological direction faster than others.

Fujitsu’s Legacy Modernisation offering allows for such technologies to be replaced or upgraded without the need for decommissioning the entire system and replacing it, which would often be an expensive, high-risk alternative.

With this in mind, Fujitsu has partnered with OpenText after undertaking significant due diligence activities to ensure that the OpenText offering is truly the best of breed alternative for migrating from IBM® Notes® applications to a Microsoft®-based alternative and Oracle Forms to Java.

The main issue to address when undertaking checks on candidate migration partners is the quality of the migrated application. There is little point in migrating a system if the resultant deliverable is difficult to maintain and develop. The OpenText option delivers high quality migrations again and again.

As OpenText’s senior Systems Integrator in the Australasian region, Fujitsu brings a critical mass of resources and experience in migration projects of many types that can be utilised to ensure that a migration project can be undertaken in the timeframe and to the quality required.

The Fujitsu Legacy Modernization team is pleased to be a partner with OpenText and we are actively engaged in rolling this offering out to the other 16 Legacy Modernization teams around the world.

Fujitsu Legacy Modernisation

About Fujitsu Australia and New Zealand

Fujitsu Australia and New Zealand is a leading service provider of business, information technology and communications solutions. As the third largest ICT Company in the Australian and New Zealand marketplace, we partner with our customers to consult, design, build, operate and support business solutions. From strategic consulting to application and infrastructure solutions and services, Fujitsu Australia and New Zealand have earned a reputation as the single supplier of choice for leading corporate and government organisations.

About Fujitsu

Fujitsu is a leading provider of IT-based business solutions for the global marketplace. With approximately 186,000 employees supporting customers in 70 countries, Fujitsu combines a worldwide corps of systems and services experts with highly reliable computing and communications products and advanced microelectronics to deliver added value to customers. For more information, please see: www.fujitsu.com or email: legacy.modernisation@au.fujitsu.com.