The Vgo Software Approach to Oracle Forms Conversions Using the Evo™ Conversion Engine

A Vgo Software Whitepaper, October 2006

Author: David Holk, Architect

Executive Summary

Vgo Software is one of only two Global Partners certified and recommended by Oracle Corporation for conversion of Forms to Java. Vgo Software uses its Evo™ conversion engine to automate moving Oracle Forms to a J2EE platform.

The Vgo Software's Forms conversion approach using Evo provides a broad range of conversion options. An application can be converted strictly function for function, or dramatic enhancements can be made, positioning a client for adoption of strategic enterprise initiatives such as service-oriented architecture or mobile computing. Evo's conversion engine drastically reduces the time and effort necessary to perform Forms conversion by combining the strength of code generation with conversion and translation features. Applications built with Evo are open, standards-based Java applications. The output from an Evo conversion is pure source code; there are no proprietary DLL's, runtime engines, translation artifacts or development environments required for the client, post-conversion.

The Challenge

Oracle Forms are very common in many organizations. It is estimated that as many as 400,000 companies use Oracle Forms in a licensed or unlicensed form. As companies move toward open Java-based architectures and support for legacy, client-server Forms applications decreases, they must migrate their applications to a current, better-supported technology.

There are options for companies to get their applications "onto the Web":

1. Forms can be migrated to 10g Web Forms
2. Forms can be converted to Oracle's JSF/ADF Faces
3. Forms can be migrated to an agnostic J2EE implementation (My Faces JSF, JSP/Struts, Swing)

Technically, each option has its pro's and con's. Vgo Software takes into consideration the client's strategic direction such as working with common procedures and conventions from other enterprise level projects, a move towards centralization of services or, minimally, the re-use of common application code and adaptation of new technology from an organizational perspective (i.e. the readiness of a client to adopt J2EE and support it comfortably going forward).

Vgo Software believes that the strategic direction of the client is equally important to technically converting code. From a Vgo Software perspective, a "straight conversion", which does not take into consideration any enterprise strategic value points, is considered a "blind conversion". In a "blind conversion", a client is left with a 1:1 conversion rate to their existing form (or derivative based on the conversion chosen) and has no real control of the source generated.

Vgo Software believes our approach of combining Evo with our Evolutions Methodology (a conversion methodology developed by Vgo Software to properly design and address conversion issues for a better all around end-product) is the best solution on the market. Vgo Software attempts to address all of the challenges clients face in this migration to create a better positioned, highly supportable, end-state application.

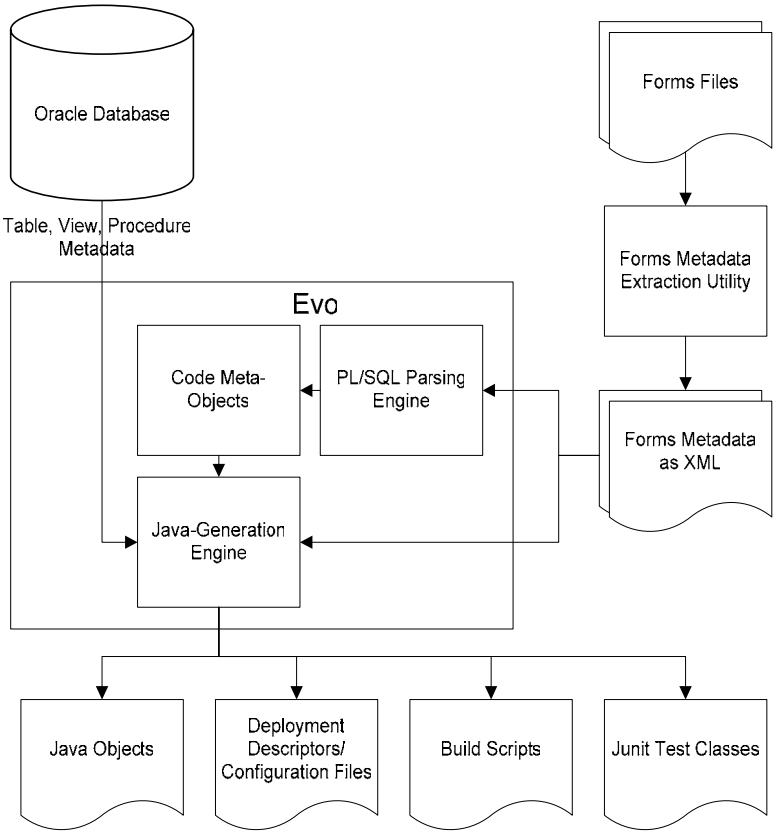
The Solution

Code Conversion

The Evo conversion engine can automate the majority of a Forms-to-Java conversion project. Evo loads in a Forms file (an XML version of a FMB) and extracts information about all the data access, UI elements, and PL/SQL code from triggers, functions and procedures. Evo utilizes a powerful PL/SQL parsing engine to translate it to Java.

During this process, Evo extracts the tables, views and stored procedures the application uses from the PL/SQL code and Forms metadata. Using this data and metadata from the database, Evo creates a data-persistence layer using the client's choice of data access frameworks. By default, Evo create a JSF client for the UI of the application.

In addition to standard browser-based clients, ADF can be used with Oracle's ITS telnet services to create character-based applications that run on handheld or vehicle mounted devices via telnet. Using Oracle ADFT & ITS, clients are afforded the only viable path for Character-based forms conversion to Java. At the time of this writing, Vgo Software is the only company in the world to offer solutions and experience using this technology.



Forms Conversion with Vgo™ Evo
Figure 1

Client Implementation Options

There are several options in getting Forms "to the Web". In this section we will discuss only options for J2EE conversions. Migration to Oracle 10g Web Forms is not discussed.

JSF/ADF Faces

Java Server Faces (JSF) is Sun's official Java framework for building user interfaces for web applications. JSF is essentially a JSP tag library along with an associated object library for maintaining state, validating user input, controlling navigation and other common web application tasks.

There is quite of bit of overlap in the services offered by JSF and Struts, although each provides functionality that the other does not. Simply stated, Struts offers a more robust MVC implementation, while JSF provides better fleshed-out UI widgets. With the ability to use 3rd party render-kits or custom-built render kits, JSF can be used as the basis for

applications that typically are not designed using web applications. JSF render kits can provide interfaces in HTML, VML (Voice Markup Language), or even Swing.

ADF Faces is a set of user interface components, including a custom render kit, created by Oracle. These user interface components are implemented as a standard JSF tag library along with all the necessary supporting classes. ADF Faces also supports partial page rendering, which allows for partial refreshes of pages. ADF Faces provides a rich library of JSF widgets.

An interesting benefit of using ADF Faces is the telnet functionality provided by ADFT/ITS (Application Development Framework for Telnet/Industrial Telnet Server). ADFT/ITS provides a facility for building telnet applications using JSF. This is an ideal option for companies that are moving forward with J2EE but still need to support legacy hardware such as handheld scanners or vehicle-mounted devices. This option is viable for any client running character-based forms and wishes to move to an enterprise-class J2EE environment.

JSF/MyFaces

The MyFaces JSF implementation is the continuation of the Oracle ADF Faces components donated to the Apache Foundation and that are now available as an Open Source product. MyFaces is still a part of Oracle ADF Faces but there is no licensing fee required by Oracle for a client to use the MyFaces components.

JSP/Struts

Java Server Pages (JSP) is a Java framework for creating dynamic content for the web, typically in the form of HTML and JavaScript. A JSP page is a text document that contains two types of text: static data, which can be expressed in any text-based format, and JSP elements, which construct dynamic content. The JSP elements that generate the dynamic content are either scriptlets, fragments of Java embedded in special tags, or JSP tags.

Struts is an open-source implementation of the Model View Controller design idiom, maintained by the Apache Software Foundation. While not a Java standard in the strict sense, its wide-spread use has made Struts an industry standard for the last several years.

The JSP/Struts client implementation option uses standard Java Server Pages and their associated beans in conjunction with the Struts framework. Among its many features, Struts provides a consistent way to map user interactions with the application to the Java classes that implement the desired behavior.

Swing

Swing is the standard Java API for building client-server applications. Though these applications typically include a thick client-side interface, new designs allow for thinner client-side applications that use web services.

Though Vgo Software supports Swing conversions, we do recommend reviewing the option of the conversion to Oracle Web Forms. In most cases, client's considering the move to Swing are doing so based on the ability of Swing to render equivalent richness of the original Forms user interface. From a UI perspective, this is the simplest of all conversions, *after* the Oracle Web Forms conversion.

If clients require that the original Forms interface be preserved, Vgo Software will likely recommend Oracle Web Form migration over a Java Swing conversion. This recommendation is based on supportability, easier learning curve for existing programming staff, and a simpler all around conversion.

Evo and Customization

No matter which architecture you choose to convert your Oracle Forms application to, Vgo Software will work with you to ensure that the converted application conforms to the standards of your organization. Vgo Software's conversion process provides the opportunity to incorporate changes upfront so that the converted code generated conforms to any existing corporate standards at the start.

If you have a proprietary J2EE framework, or have adopted a third party's, Vgo Software can modify its Evo Constructors so that the conversion conforms to the pre-existing application architecture.

This is an important differentiator, as the converted application can now be created in patterns matching other enterprise-wide J2EE implementations or work with existing J2EE frameworks.

Data Persistence Layer Implementation Options

The implementation options explained in this section are all referred to as "default" Evo implementations because that is exactly what they are. The greatest benefit of the Evo conversion engine is the way it allows for easy modifications to adapt to the user's environment. The persistence layers presented in this section are what the user would generate for a persistence layer if no changes were made to the Evo constructor responsible for converting the application.

EJB/CMP

The default Evo EJB/CMP implementation provides the features of Enterprise Java Beans; integrated transaction management, built-in scalability, and sharing of the data persistence layer across applications. Using the EJB/CMP layer also allows for transactions that span disparate data sources without the need for customized Java Transaction code.

EJB/BMP

The default Evo EJB/BMP implementation allows for the best features of both the DAO and EJB/CMP layer. While more complex than the straight forward DAO layer, this

option allows the container to manage transactions across data sources and also provides the ability to easily share these objects across the enterprise.

JDBC

The default Evo JDBC implementation is a lightweight data persistence framework that uses plain-old Java objects (POJO's) along with JDBC as opposed to EJB's. This data layer option is less complex, more easily maintained and performs better than its EJB counterpart. It is the preferred design pattern when there is no need to share data objects with other applications and there are no transactions that span separate data sources.

ADF Business Components

The Oracle Application Development Framework provides a Business Components framework for persistence. Used in conjunction with the ADF Model Layer and ADF Faces, together these technologies provide an easy way to develop and maintain J2EE applications. Applications developed in these frameworks will be especially easy to maintain in Oracle's JDeveloper IDE. Oracle aims to provide an easy transition for Forms developers to ease into J2EE and the ADF frameworks are the realization of that goal.

True Evolution – Object Reuse

One of the biggest drawbacks of automated conversion of client/server applications to J2EE is that it has been difficult to automate object re-use in the conversion itself. With Evo, users can associate Oracle Forms objects (triggers, program units, etc.) to parent objects through the Design stage of the tool. This allows users to consolidate redundant Forms code into callable objects which are in turn requested from the converted application. In essence, "convert once, call many times" becomes reality.

Evo provides insight into re-use through visibility in the Analysis stage of the tool. Users can discover which Forms components should be consolidated from both a Program level (triggers, program units, etc.) or a Data Access level (queries within the Forms from multiple components). Once it has been determined that component's can be consolidated, a Parent object can be created and the components associated to it. Child objects inherit from the Parent and are placed in their appropriate calling positions in the generated code.

Embracing object re-use in this conversion allows for a highly maintainable and easy to understand Java application. The ability to consolidate components to single instances also makes it much easier to incorporate a converted client/server application into a SOA architecture.

Manual Conversion Tasks

A conversion effort as dramatic as the move from Oracle Forms to J2EE requires some manual development work. These manual tasks address complexities in the conversion, but also allow flexibility in the manner with which certain program units are converted.

These tasks might include:

- Modifying events that do not have a direct equivalent in the target paradigm to the events that are available and a best fit.
- Finalizing or modifying the layout of the generated UI.
- Clean up any PL/SQL code that did not optimally convert (*Evo continues to improve its PL/SQL conversion; at time of writing approximately 80% of PL/SQL in a form is converted cleanly).

Other manual conversion tasks that may occur depend upon the conversion process chosen by the customer. A straight forward conversion will create a functional Java-based application in the chosen paradigm, but may not be as optimal as if the application had been designed from the start with the target platform in mind.

Vgo Software can assist by modifying Evo's generation techniques to provide a base that is as close to the desired design as possible in generated code, but truly architecting a solution that conforms to all the standards and practices in an organization requires more custom work. This is where Vgo Software **Evolutions Methodology** fits in Vgo Software works with the client's architectural team to determine optimal design patterns that match the organizations, facilitate use case and sequence design meetings for changes required when adopting the web paradigm, customize transaction and security models, and raising enterprise service-level components from the existing Forms base so that they can be used in an SOA going forward.

Some target platforms are better suited for direct conversion with minimal manual effort. A Swing conversion can be very straightforward and may not require as much customization as a pure web application in which the workflow itself may be redesigned to better fit into the web paradigm (see "Client Implementation Options" earlier in this document).

With any conversion, Evo leaves the original code behind, but commented out (this is an option in Evo), in the converted equivalent. This makes any manual activities very easy to deal with. We also recognize that we can not make architectural decisions for you automatically. In fact, no automation software should have that responsibility. Therefore, it is simple to read original code and architect the right component for your new environment. Finally, leaving the original code in the converted application makes the new application easier to support by Oracle developers learning Java; they can see and understand the original PL/SQL and see its equivalent clearly spelled out in Java.

Evo is the only tool on the market that offers customization options that will optimize the output to be aligned with a strategic direction. The Vgo Software Evolutions Methodology allows this optimization to occur in a repeatable, process-driven, manner.

Why Vgo Software and Evo?

Evolutions Methodology: The Flexible Conversion Methodology

The move from Oracle Forms to Java is a significant change. Making the move can provide an opportunity to implement enhancements or remove unused functionality.

Going to a true, browser-based, web application such as My Faces JSF, JSP/Struts or ADF may require re-engineering and re-architecting the way original client/server applications behave. If this is not understood by a client, then migrating (upgrading) Forms is a likely path.

Vgo Software recommends that *all* potential conversion candidates go through some form of conversion design prior to undertaking the conversion itself. Making the jump from Forms to Java is a big step with a lot of opportunity to be positioned strategically at the end, specifically in the areas of re-use and positioning for SOA (service oriented architecture).

Vgo Software has recognized this as a formal series of steps in the process and has created its Vgo Software Evolution Methodology. The Methodology walks through the entire conversion process, including re-architecting and conversion engineering early in a conversion project. It is complete with formal artifacts, "lessons" on how to conduct certain sessions, lessons-learned through a conversion and more.

Creating a conversion architecture and planning on re-architecture are the appropriate measures to take in this activity. "Blind conversions", those that claim "near 100%" conversion rates, simply take Forms code and attempt to make a direct translation to Java. In cases where this works (typically Swing provides the highest conversion rates as it is essentially a thick client adaptation of Java serviced through an applet or application), the code is unreadable, unmanageable and the conversion itself leaves the client in no better position than simply migrating the application to Oracle Web Forms.

Taking the time to plan on changes to process flow, transaction management, security architecture, end-state positioning and application packaging, among others, simply leaves a client in the best possible condition to support, enhance and manage their application going forward.

Expertise

The Vgo Software team includes top-tier Java and Oracle talent. Vgo Software has a unique combination of in-house Oracle Forms and Java expertise that, coupled with the Evo conversion engine, can perform Forms conversions in a fraction of the time of a manual rewrite while combining flexibility in pre-conversion design (Vgo Software Evolution Methodology) with automated conversion (via Evo).

Technical expertise is not the only factor in this conversion. Quality is equally important. Vgo Software follows a rigorous testing and quality assurance methodology to ensure that the conversion has been completed with a high degree of fidelity.

Vgo Software has been selected as 1 of only 2 partners certified by Oracle product development to do this kind of work.

Vgo Software also has a dedicated Information Architecture practice, through its parent company, NEOS, that focuses on data and object model design and implementation. The combination of both practices can bridge the gap of migrating only the application architecture; there is expertise in-house to help assist with reviewing the information architecture for the ultimate end-state conversion.

Conclusion

Automated conversions with the Evo engine can bring legacy Forms applications forward into the Java world at considerably less expense than rewriting the code. This open approach provides flexibility in the conversion process as well as industry standard, readily maintainable code.

Vgo Software believes this is the best approach to truly critical, enterprise-class, applications. The approach combines the best aspects of automation with engineering discipline to provide an open, maintainable, robust end-state application.

The VGO SOFTWARE Approach to Oracle Forms Conversions Using the Evo Conversion Engine
October 2006
Author: David Holk
VGO SOFTWARE Software, Inc.
160 Chapel Road, Suite 103
Manchester, CT 06040
Phone: + 1.860.533.9383
Fax: + 1.860.533.9743
www.Vgo Software software.com
www.Vgo Software llc.com