Migrating Oracle Forms Applications to J2EE & SOA with the NewCode Competitive Migrator 3.0 Edition for IBM Rational Software Development Platform.

NewCode

MODERNIZATION TECHNOLOGY

MEMBER OMG OBJECT MANAGEMENT GROUP

Ready for IBM Rational. software

Migrating Oracle Forms Applications to J2EE & SOA with the Competitive Migrator 3.0 Edition for IBM Rational Software Development Platform

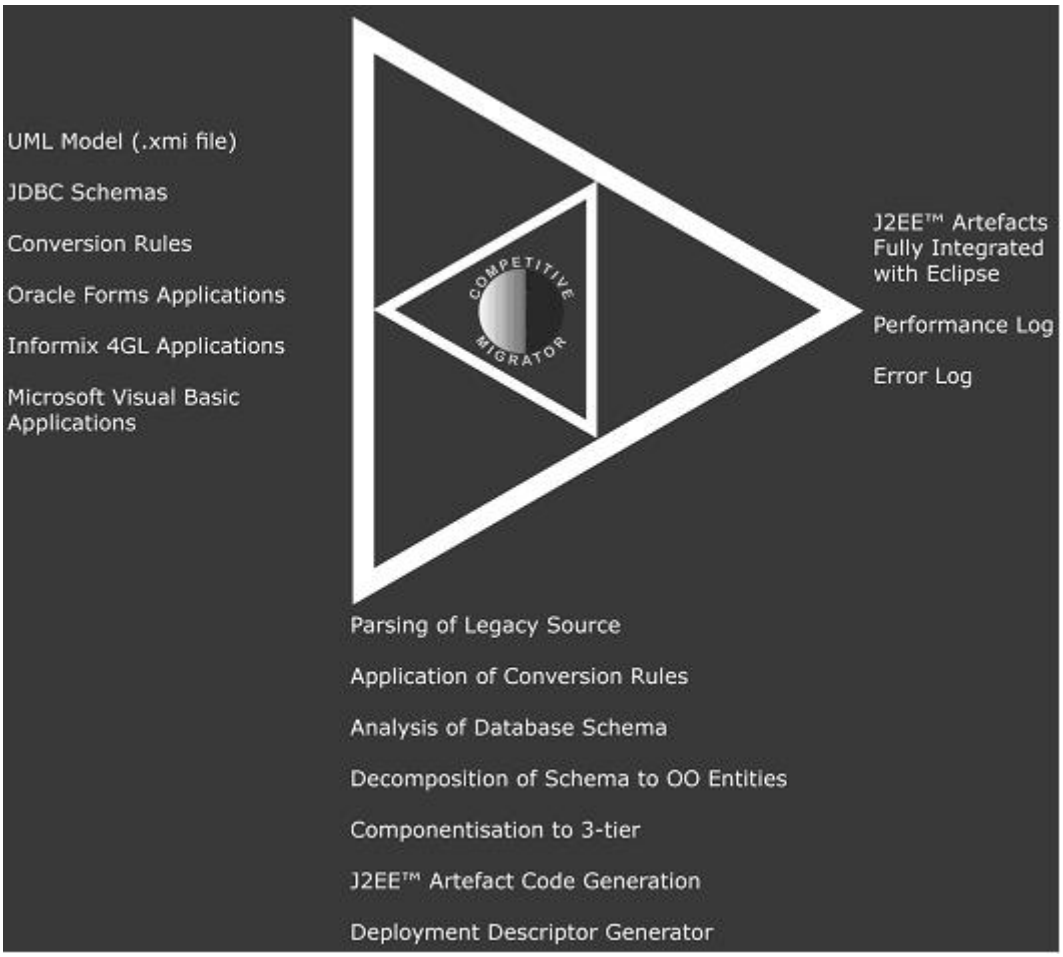
June 2007
Part Number E10676

Introduction

This document outlines the migration processes, architectural features and deployment options when using the NewCode Competitive Migrator 3.0 to migrate Oracle Forms applications to n-tier J2EE.

The **Competitive Migrator** from NewCode is a model-driven software toolset that provides a series of adaptors for modernizing Oracle Forms legacy applications into J2EE–SOA applications and UML models. The toolset runs as a plug-in to the IBM Rational Application Developer for WebSphere Software. Re-engineering can take place on the NewCode model to remove redundant application elements and include new functionality, such as a layer of Web Services. Code can then be generated for deployment and the model can be converted into UML 2 models for further development within the IBM Rational Software Architect modelling environment.

Uniquely, with the Competitive Migrator it is possible to re-architect the solution from a two-tier, client server application built on a structured methodology to an n-tier J2EE application built on an object-oriented methodology. This is because modernization with NewCode is a multi-step approach that enables the user to create and work with application models and UML2 models in order to decompose the business logic into object-oriented entities. We recognize that no tool could automate such a process so we provided a patented process with modelling capability to allow users to complete this element of the migration work rapidly and also to facilitate acceptance testing for the customer.



With NewCode you can implement a recognized, Service Oriented Architecture, including conversational Web services, page flows, business logic control layers, application, Web and Enterprise Java Bean projects, all without necessarily having to write a single line of code. The solution for the customer is a functionally equivalent J2EE1.3/4 application ready for deployment on application servers, such as IBM WebSphere Application Server together with other J2EE projects and artefacts. The new application code is generated into the Eclipse-based IBM Rational Software Development Platform environment, for ongoing development and maintenance.

STANDARDS BASED

All application components generated by the NewCode Competitive Migrator are built on open standards, and the migrated code complies with open standards. You can generate, build and deploy state-of-the-art, server-side applications utilizing Service Oriented Architecture (SOA), that meet the J2EE1.3/4, Java Servlet 2.3, JavaServer Pages (JSP) 1.2, Enterprise JavaBeans (EJB) 2.0 specifications and beyond., to open standards-based web application servers such a IBM WebSphere Application Server.

THE BENEFITS OF MIGRATING WITH THE NewCode COMPETITIVE MIGRATOR Edition for IBM Rational Software Development Platform

You can achieve a high level of business value from IBM/ NewCode's unique approach to the migration and accelerated development of J2EE-compliant applications.

- Remove your department from dependencies on proprietary tools, middleware runtimes and databases.
- Leverage your existing investment in database and application logic in the migration
- Realize productivity from a modern pattern-based, model-driven development technique over manual coding of Java artifacts.
- There is an extensive library of preferences that can be set to provide a corporate 'look and feel' across all generated applications.
- Applications can be reshaped by quickly changing parameters and regenerating without recourse to manual coding
- Potentially achieve an 85% reduction in the time required to manually perform the same migration analysis of Oracle Developer modules (may virtually eliminate expensive pre-analysis effort to 15%)
- Implement a recognized, Service Oriented Architecture (SOA), including conversational Web services, page flows, business logic control layers, application, Web and Enterprise Java Bean projects, all without necessarily having to write a single line of code.
- Remove the risk of error associated with tedious "eye-balling" of legacy source code and other security risks associated with manual coding
- You will always have full ownership rights of the IP from applications developed and deployed using the NewCode tools
- Automatically enforce J2EE1.3, 1.4 standards and beyond
- Dramatically reduce annual spending on expensive and scarce development and testing resources.
- Drive productivity and regulatory compliance through ongoing use of the IBM Rational Software Development Platform
- The NewCode Competitive Migrator is delivered as an integrated plug-in to IBM Rational Software Architect and the Rational Application Developer and enables exchanges of artefacts transparently with the full range of Rational Application Development Life Cycle solutions.

System Prerequisites. The NewCode Competitive Migrator 3.0 Plug-in requires the following minimum system configuration:

Component	Recommended Requirement
Processor	Intel Pentium® III (800 MHz or higher recommended) AMD Dual Core
RAM	RAD and NewCode min 1G
IDE	IBM Rational Application Developer for IBM WebSphere Server V6.0 IBM Rational Software Architect V6.0
Application Server	IBM WebSphere Application Server V5.+
Operating System	Windows 2000, or Windows XP RedHat Enterprise Linux Workstation V3
Browser	Microsoft Internet Explorer 5.5, Netscape Navigator 4.07 or higher
J2EE™	J2EE™ 1.3, 1.4
Oracle Forms	The NewCode Conversion Wizard supports Oracle applications written in Oracle Developer 2000
JDBC Driver	Oracle JDBC 2.+ (AIX, HP-UX, LINUX, SOLARIS, WINDOWS) or equivalent

Product Overview

With NewCode, the customer is not simply left with line-by-line migrated Java code. The customer is empowered to re-model, re-engineer, re-generate and re-deploy the migrated applications in order to take advantage of n-tier J2EE and SOA.

The Process Inputs

1. A Base Model (a NewCode model template that can be user defined containing default model objects)
2. Oracle Forms files, libraries and PL/SQL applications
3. The target environment. The user can choose an application server, including IBM WebSphere Server
4. A UML2 (or XMI) schema representing Reusable objects (modelled using Rational Software Architect). (optional)
5. An existing JDBC compatible database schema (optional)

Setting Preferences
With the NewCode Preferences Module, the user can implement a range of application level preferences which apply to all migrated applications, including:

- Base model choice
- Base model configuration
- Default model element properties
- Default Language
- Default IDE
- Generated documentation format
- Deployment location
- Server home
- JDBC drivers
- Library Modules
- Logging
- Migration Adaptors
- UML Stereotypes
- Look and Feel Templates
- NewCode template tags

Running the Migration Wizard and Oracle Forms Adaptor
The wizard collects information from the user and runs the Oracle Forms Adaptor which supports Oracle Forms files (see later section on Oracle Forms Adaptor) and is capable of analysing Oracle application syntax and applying conversion rules to it. The wizard performs the following automated, conversion tasks:

- Establishes a database connection via JDBC
- Runs the Oracle Forms Adaptor which processes the entire set of Forms and libraries including PL/SQL source code.
- Identifies and skips dead code
- Restructures unstructured code
- Converts up to 90% of the Oracle Forms system to standard Java 2 code using the Oracle JDAPI and an AST (Abstract syntax tree) based on a purpose designed ANTLR grammar.
- Decomposition of the schema into OO entities
- Generates data access code via JDBC
- Generates code documentation
- Preserves in the migrated Java code the name of the original Oracle command structures, variable names, routine names and comments
- Generates a performance and error log

The wizard assimilates the results of the conversion into a **NewCode model** representing an n-tier J2EE SOA application. Any legacy application code that could not be readily transformed into model elements is placed in Code Block files, for manual re-engineering. The code block files are linked to Code Block model elements and contain Java code converted from the legacy source.

Re-engineering with the NewCode Model
The NewCode model is built on the Eclipse Modelling Framework (EMF), as is the Rational Software Development Platform, and provides a powerful means of both re-engineering and maintaining applications post migration. The model includes a full range of user interface and services model elements including Page Flows, JSP, Tag Libraries, Servlets, Code Blocks, Web Services, EJB and JDBC controls. Entire applications can be represented in the model. Following migration the user can re-engineer their application by adding, removing or updating model elements and their properties, and including new model elements from existing components in Rational Application Developer. Because the Competitive Migrator is built on the Application Generator, new application components can be easily modelled.

Re-engineering with Rational Software Architect

The NewCode model can be converted to and from UML 2 enabling IBM Rational Software Architect to be used for further development of the migrated application through UML modelling.

Code Generation

Following re-engineering (and at any time post-migration) application code can be generated from the model using the NewCode Code Generation Adaptor plug-in and deployed to Rational Application Developer 's supported deployment platforms.*

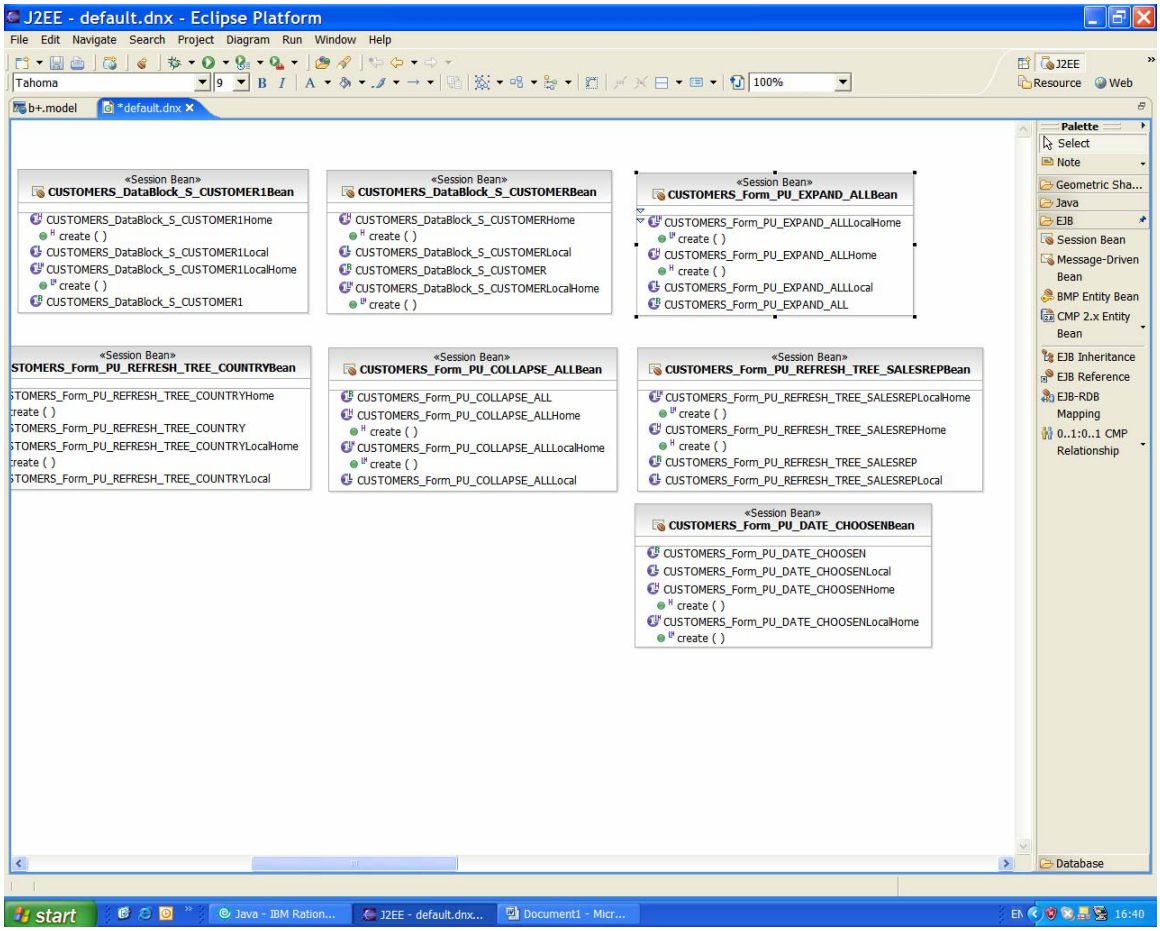
* IBM Rational and its business partners provide deployment support for WebSphere Application Server, as well as WebLogic, Oracle and JBOSS servers.

Code Compliance Testing with the Sun AVK

The NewCode products can be integrated with Sun Microsystems' Application Verification Kit (AVK), which provides a report on Java standards compliance and best practise on the generated code, every time the code is generated.

Maintaining Applications Post-Migration

Customers that license the Rational Software Development Platform products and the NewCode Application Generator have choices for maintaining their applications post-migration. With the NewCode Application Generator, users can create new model elements, remove elements created by adaptors, drag and drop objects across application tiers, change look and feel templates and amend element properties such as target deployment platform.



Competitive Migrator generated Session Bean artefacts in IBM Rational Software Architect.

IMPLEMENTING AN SOA

The Competitive Migrator provides organizations with the ability to leverage existing assets, legacy applications and databases, and make them part of the overall enterprise solution, by migrating and extending them into the Service Oriented Architecture (SOA), rather than by simply performing a line-by-line migration to Java code. Following the running of migration adaptors and re-engineering, the NewCode code generator produces an application with its logical processes mapped to services, which potentially allows them to operate independently, interacting with other systems using well-defined contracts.
The Competitive Migrator virtually eliminates expensive pre-analysis effort to 15%, an 85% reduction in the time required to manually perform the same analysis of Oracle Developer modules and, without the risk of error associated with tedious “eye-balling” of legacy source code and other security risks associated with manual coding.

THE GENERATED COMPONENTS

Industry Standard Architecture
The NewCode Competitive Migrator is shipped with industry standard Application Models, Patterns and application frameworks (including the Model-View-Controller (MVC) design pattern, Sun’s core J2EE patterns e.g. Sun’s Core J2EE Service Locator pattern and Apache Struts for the user interface layer). The NewCode Code generators use patterns that are compliant with Sun’s J2EE Pattern Catalogue.

Enterprise Application Components
The NewCode Competitive Migrator is capable of generating Enterprise Application projects within Rational Application Developer, containing files, folders and references to the resources needed for enterprise applications. The modules in an Enterprise Application project are mapped to other J2EE™ projects and mapping information is stored in metadata files within the Enterprise Application project. Metadata files are generated for exporting the project to an EAR file, for deployment and running the project on the server.

- Wizards generate resource structures that comply with the Servlet and J2EE™ specifications
- Generation of an application deployment descriptor (application.xml) file

Web Applications
The NewCode Competitive Migrator provides the capability necessary to generate Web applications as defined in the Sun Microsystems Java™ Servlet 2.3 Specification and the Sun Microsystems JSP 1.2 Specification. Web applications include static Web pages, JavaServer Pages (JSPs), Java Servlets, an XML deployment descriptor (web.xml), and other Web resources.

- J2EE Web project generation, using the J2EE-defined hierarchy
- Generation of a Web application deployment descriptor (web.xml) file
- JSP and HTML file generation and validation
- Custom JSP tags (tag library) support, based on the Sun Microsystems JSP 1.2+ Specification
- JSP front-end based on the Struts framework
- Support for customized 'look and feel'
- Cascading Style Sheet (CSS) support
- Link generation with parameters
- Servlet generation and adding servlet mappings to the Web deployment descriptor files
- Generation of Web pages from templates including splash, menu framesets, read-only mode, update mode, master-detail, tabular lists and search pages.

Java Page Flows
During the migration process, the NewCode Competitive Migrator transforms the Oracle Forms form triggers into page flow actions and generates a set of page flow model elements representing transactions. Built on the Struts framework, page flows represent the main part of a web application software engineering model that enables architects and developers to implement web applications by using easy-to-understand, easy-to-maintain patterns.

The concepts behind page flows - separation of business logic and presentation, and centralized navigational control - reflect a design pattern: Model-View-Controller (MVC).
Page flows allow the separation of the user interface code from navigational control and other business logic. The separation of presentation and business logic offers a big advantage to development teams.

Web Services
The NewCode model utilises XML and HTTP Web Service technologies that have become most widely adopted for service interaction and service description. WSDL (Web Services Description Language)

specifies how to describe a service using a special XML document format. A WSDL document is the contract for the service. For any client to understand how a Web service operates, the service must describe itself using WSDL. The NewCode Competitive Migrator uses WSDL as its universal service contract language. WSDL is used for defining the service contract regardless of the underlying transport mechanism being used for SOAP messaging. With NewCode, Web Services can easily exchange messages using other protocols than HTTP, such as SMTP, simply by changing properties on the Web Service objects in the model. This is an approach that provides a powerful and extensible contract language that is used to define the Web service components and provides a standard's based approach to exposing database and legacy systems as Web Services.

If for example, the user has selected IBM WebSphere as the target environment, the NewCode Code Generator generates service contracts compatible with WebSphere technologies. The generated Web service communicates with clients and resources over standard protocols such as HTTP by exchanging XML messages. WebSphere Server on which the web service is deployed is responsible for routing incoming XML messages to the generated Web service code. The Web Service exports a WSDL file to describe its interface and providing access to the service.

EJB Components

The NewCode Competitive Migrator provides the capability to develop and generate enterprise beans that conform to the distributed component architecture defined in the Sun Microsystems Enterprise JavaBeans (EJB) specification. Descriptions of EJB2.+ technology are available from the following Web page: http://java.sun.com/products/ejb/docs.html#specs

- J2EE EJB project generation, using the J2EE-defined hierarchy
- Generation of enterprise beans and access beans
- Generation of entity beans with bean-managed persistence
- Generation of entity beans with container-managed data persistence
- Generation of data access objects
- Generation of stateless session beans
- Generation of back-end mapping for CMP beans
- Generation of local entities
- Enterprise beans are generated for specification compliance

No Run Time Components

Applications migrated with NewCode do not require a proprietary J2EE framework and do not have vendor-specific, run time dependencies. They can be deployed natively to IBM WebSphere Servers or other J2EE environments. All Java artefacts are generated to be compatible with any other Rational Application Developer open application components, as if they had been coded manually.

The NewCode Oracle Forms Adaptor

The NewCode Oracle Forms Adaptor understands and converts all commands and data types with a few exceptions.

Introduction:

The legacy Oracle Forms and Designer environment is complex with a proprietary set of capabilities that don't fit well with UML and Java at present. Forms allowed users to map database tables using datablocks and provided a framework that took care of low level details such as executing SQL statements, fetching data, locking rows and detecting lost updates. In a J2EE framework, entity EJB's are mapped to database tables and are responsible for transactions and data persistence. Database connection pooling and the coordination of transaction commits and rollbacks are the responsibility of the application server.

Unlike Oracle Forms, In J2EE the user interface is physically separated from the back-end data which simplifies reuse of data components and enables business logic to be held separate from the user interface.

The NewCode Oracle Forms Adaptor:

The NewCode Oracle Forms Adaptor undertakes the preliminary work in the migration of an Oracle Forms application to J2EE architecture. The adaptor will analyse the source application and construct a NewCode model that can be used to re-engineer a migration solution. The generated model reduces the process of migration to a single re-engineering phase during which the user can customise the model to suit the individual requirements of the new application. The user can also, if desired, add new functionality above and beyond the functionality of the original Oracle Forms application.

All Oracle Forms code and data types are automatically converted, with a few exceptions. Applications developed in Oracle Forms Developer consist of a set of forms and libraries. A language parser and lexical analyser are used together with an object parser to map the inheritances and relationships of the business logic framework. The presentation logic, validation rules, business logic are extracted and transformed into an intermediate model, built on the Eclipse Modelling Framework (EMF). The PL/SQL code and data access is converted into Java methods on the presentation and business tiers. Model elements represent application components and the user has significant control over the J2EE components used in the presentation layer, controller layer and model layer.

How the J2EE design maps to the original Forms

View: this is the application user interface which maps to the Forms applet for rendering the UI. In J2EE the view is typically presented as Java Server Pages.

Controller: this handles the application flow and co-ordinates the activity between the Model and View layers. In Forms this maps to the navigation triggers and Forms code. In J2EE the Controller is typically implemented by the Struts framework.

Model: this handles interaction with data sources and runs the business logic. This maps to the Forms data block which implements block level properties, data related logic and data persistence. In J2EE a number of business services are available including Web Services, EJB, JDBC, Java beans and others.

Mapping of Oracle Developer File types

Supported Oracle Component Type	Oracle Developer Sub Component	J2EE Application Component
Form Modules (.fmb)	Form Level Triggers	Form Bean validation Action Code Blocks
	Alerts	Application Resources File
	Data Blocks	Stateless Session EJB or JDBC Controls
	Data Block Queries	EJB or JDBC Methods and Java Controls
	Data Block Triggers	EJB or JDBC Methods and Code Blocks
	Data Block Items	JSP Field Controls, Form Bean Fields:

		1. A visible UI Control on an Oracle Form becomes a UI Control. The selection of control depends on the tag library and UI technology the user chooses e.g HTML or JSF. 2. A Forms 'Push Button' is mapped to a Button, Button with Image, Submit Button or Reset Button. 3. A Forms Image is mapped to an Image in the model 4. A Forms Drop down menu (Poplist) is mapped to a Select control. 5. A Forms editable text area of size > 256 is mapped to a Text Area. 6. A Forms editable text area of size < 256 is mapped to a Text Box
	Data Block Item Triggers	Action Code Blocks
	Canvases	Page flow (may be shared with other Forms), JSP, HTML Forms and Layout Sections
	Canvas Graphic Frames	Layout Sub-section headers
	Canvas Graphic Text	HTML/JSF Labels
	Editors	HTML/JSF text area
	LOVs	Read-only JSP, Table, Parameters, Form Bean field, Action Method Reference (to Web Services/Model layer)
	Object Groups	Java Controls and various
	Parameters	Session Attributes
	Popup Menus	JSP URL / Links
	Program Units	Stateless Session EJB or JDBC Controls
	Program Unit Packages	Multiple EJB or JDBC Methods with parameters, return type, return mapping, exceptions and Java Controls
	Program Unit Procedures	EJB or JDBC Method with parameters exceptions and Java Controls
	Program Unit Functions	EJB or JDBC Methods with parameters, return type, return mapping, exceptions and Java Controls
	Property Classes	HTML or JSF control with CSS and / or model properties
	Record Groups	Stateless Session EJB or JDBC Controls with EJB or JDBC Methods with parameters, return type, return mapping, exceptions and Java Controls. The method contains the SQL query used to populate the values in the Form Bean and JSP representing the LOV.
	Visual Attributes	CSS and / or model properties
	Windows	JSP (or portlet)
Menu Modules (.mmb)	Menus	JSP URL / Links

	Program Units	Stateless Session EJB or JDBC Controls
PL/SQL Libraries (.pll)	Program Units	Stateless Session EJB or JDBC Controls
Object Libraries (.olb)	Built Ins	Java Controls and various
Reports (.rdf via .xml)	Data Model	Stateless Session EJB or JDBC Controls with EJB or JDBC Methods with parameters, return type, return mapping, exceptions and Java Controls
	Paper Layout	JSP with Layout Sections, HTML tables, HTML / JSF Control Fields and Table Controls
	Report Triggers	Action Code Blocks
	Program Units	Stateless Session EJB or JDBC Controls with EJB or JDBC Methods with parameters, return type, return mapping, exceptions and Java Controls

Support for all versions of Oracle Forms and Reports

The adaptor supports all versions of Oracle Forms and Reports

Database Packages/Procedures/Triggers

The adaptor does not migrate the database schema to other databases, however it is capable of migrating PL/SQL procedures and packages stored in the database.

Support for Arabic, Hebrew and bi-directional languages

The adaptor fully supports migration of Arabic and Hebrew for editing directly in the intermediate model and in the generated application.

Dependencies/Requirements:

1. Oracle Forms 10g must be installed, as this migration adaptor uses the Oracle JDAPI to access the proprietary binary format of Oracle Forms source files. This API sits on and uses the Oracle Forms 10g application. The JDAPI jar file must be added to the CLASSPATH or the jar file imported. This jar file comes bundled with Oracle Forms 10g.
2. Although the adaptor does not migrate databases, a JDBC connection to you database is required to import the schema into the intermediate model so that the model wizards can use schema information such as table field data types and sequences.

Assumptions:

1. All input files to be migrated are put in the same directory.
2. All Oracle Forms are assumed to be updateable.

Locations of critical logic in an Oracle Forms Application:

The main Oracle Forms entities considered to hold important business logic include, but are not limited to the following.

1. The Program Units in the PLL files used in the application (Attached Libraries)
2. The Program Units in the FMB files.
3. The Program Units in the MMB files used in the application
4. The 'Record Group Query' held in the Record Groups.
5. The WHEN-NEW-FORM-INSTANCE Triggers at Form Level.
6. The PRE-QUERY and POST-QUERY Triggers at Block Level.
7. The WHEN-BUTTON-PRESSED Triggers at Item level.

8. The Parameters at Form Level
9. The stored procedures in the Oracle Database which are used in the Oracle Forms Application
10. The String properties of the Data Blocks esp. Default WHERE and ORDER BY clauses.
11. The String/integer/boolean properties of the Items
12. The String/integer/boolean properties of the Oracle Forms Form Modules
13. The String/integer/boolean properties of the LOVs
14. The String/integer/boolean properties of the Record Groups

Adaptor Notes:

The Oracle Adaptor uses the Oracle JDAPI to process the source files. This API is used to build the NewCode model and populate the properties of the model elements. In the case of Reports, the user must first convert their Report to an .xml file using the Oracle Reports file conversion utility.

The Adaptor migrates the PL/SQL code to Java by converting this PL/SQL code to an AST (Abstract syntax tree) based on a purpose designed ANTLR grammar. A Tree walker then walks this AST and converts it into a combination of model elements and Java code blocks, representing the equivalent functionality. Any errors when parsing the syntax of the PL/SQL code will be written to the NewCode log file.

The UI controls on the Oracle Form are analyzed and a JSP alternative is chosen and added to the JSP. Names and labels are migrated consistently allowing easy identification on the model for re-engineering. The model supports text from a wide variety of languages including Arabic and Hebrew.

Program flows are analysed to construct pageflows. For example, a form procedure may reference an object method that in turn loads another form. In this case the two forms are within the same pageflow and as such are added to the model. One form bean is created per pageflow.

A Stateless session bean or JDBC Control will be created for each Program Unit (procedure/function/package) in each Oracle Form. This bean will house procedures and functions from the form. The procedures and functions will be migrated as EJB Methods or Service Functions. A Stateless session bean or JDBC Control will be created for each Program Unit in each PLL file also. As in the case of Form program units the procedures and functions will be migrated as EJB Methods or Service Functions.

The user can select base models that can contain ready made application model elements. In addition the NewCode tool has a wide variety of configuration options for example enabling the adaptor to automatically create a layer of Web Services or to choose from a list of target application servers.

A Java Control is also created for every table accessed by the Oracle Forms application. An Entity Bean or JDBCControl is added to the model for every Database Data block in each Oracle Form.

Actionable controls such as buttons will have an Action model object created and linked to. A Method Reference within an Action is used to link a Service Function or EJB Method to a control with that Action. For example a button click will invoke an Action, this Action has a Method Reference that points to an EJB Method, the EJB Method is thus invoked.

The adaptor migrates annotation and comments within the PL/SQL code and properties into Java annotations.

The PL/SQL to Java migrator will not convert PL/SQL code that does not compile.

Solution Deliverables:

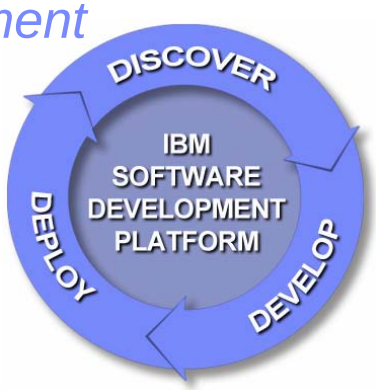
- J2EE-SOA Application model built on the Eclipse Modelling Framework (EMF)
- J2EE Architectural framework pattern
- PL/SQL Migration to Java
- Code blocks containing migrated business logic
- Transactional Pageflows
- Struts or JSF controller layer
- Web style GUI templates for GUI enhancement
- Mouse Navigation Support
- Code Optimization
- Sun AVK ANT script for J2EE code testing
- LOV enhancements
- Generated Documentation
- Export to UML2
- ANT scripts for creation of .EAR and data source file for application server deployment

The NewCode Oracle Adaptor does not create any run-times or run-time dependencies.

IBM Rational provides tools for architecture, design modeling, construction, model-driven development, rapid application development (RAD), component testing, and runtime analysis activities.

These tools help developers maximize their productivity when building business applications, software products and systems, and embedded systems and devices. IBM design and construction tools are an integral part of the IBM Rational Software Development Platform, a complete and modular platform for teams who build business applications, embedded systems, and software product

The IBM Rational Software Development Platform Summary



The most complete, open, modular, and proven development solution enhanced by “Ready for IBM Rational software” business partner extensions

The *only* solution that enables business-driven development by unifying business, operations and development and compliance processes

The *only* solution with a comprehensive strategy for improving application quality and availability post-deployment

Government industry compliance regulations are significantly increasing Information Technology operational costs today. Choosing the right software development platform can help you enforce business controls and meet mandated policies.

IBM Rational Software Development Platform automates and accelerates development of auditable, traceable and verifiable business applications. It reduces budget overruns and provides an operational advantage with better understanding and control of your department processes. The Rational solution for compliance-driven development, offers the most complete, open, modular, and proven development solution in the industry.

Business transformation requires both knowledge of existing business processes and the ability to visualize alternatives. The IBM /NewCode solution enable you to capture current business applications and activities and workflows and simulate alternative scenarios to uncover business opportunities. Once an opportunity has been identified, IBM Rational solutions help you analyze business and technology requirements, perform impact analysis on existing systems and scope quality projects appropriately.

Migration Project Offering

The legacy transformation process can be a cost-effective way of leveraging existing investments whilst reducing operational costs and accessing the capabilities of the latest application server technologies.

IBM and NewCode Services recommend a stage-by-stage approach to migration and transformation projects which combines the automation potential of the NewCode Competitive MigratorEdition for IBM Rational Software Development Platform toolset with the experience of our professional consultants or systems integrations:

The Rational/NewCode Solution may include:

- Project Assessment Services
  - Migration project review and analysis
- Application Development and Migration Software licenses
- Education and Training
- Custom Services
  - Pilot project
  - Migration phase project
  - Business integration and re-engineering project
  - Operational and deployment services project
  - etc

NewCode
MODERNIZATION TECHNOLOGY

E-Mail: info@newcode.com
Web: www.newcode.com