Forward and Reverse Engineering of Code with Rational Rose and VisualAge for Java*

by **Shawn Keller**
Delivery Leader
WaveBend Solutions

Having the ability to forward engineer, reverse engineer, or roundtrip engineer between design tools and Integrated Development Environments (IDE) offers software engineering teams powerful options in the evolution of component-based development efforts. Rational and IBM have created an effective integration between their design and implementation tools through a connector called the RoseLink Plug-In.

There are three scenarios in which using this integration point is effective:

- **Forward Engineering** -- In an instance where much thought and time has been placed into the design of the model, including modeling detailed subsystems, packaging and class structures, the architect or designer is able to forward engineer the model(s) into Java code having all of the package and class elements and relationships realized. In this case, the model is moved forward to code and the implementer(s) continues on with the generated Java class skeletons, adding business logic and building upon the superstructure designed by the architect.

- **Reverse Engineering** -- Many of us have experienced the situation of being placed on a team or engaged with a new client where the effort has no design, no modeling efforts or no clear understanding of the architecture of an existing system. Perhaps the system was built in haste or a prototype was "band-aided" into a production system. Either way, the role as an architect or implementer can be made easier if an understanding of the underlying architecture is known. By reverse engineering Java packages into Rational Rose®, a visual perspective of the system can be gained.

- **Roundtrip Engineering** -- Having experienced this personally in a rapid development effort, roundtrip engineering allows an architect
to develop the base class structure from a higher level, perhaps not having the time to design all of the properties of the classes. Having built the classes and their relationships to each other, the architect can leave the more detailed element definition to the implementer. Roundtrip engineering can keep both the design model and Java code in synchronization. The architect will forward engineer the model to code, the implementer makes additions or validated changes to the structure and reverse engineers the code back to the same Rose model. The architect can then visualize the implementer's changes in the model. These roundtripping efforts can effectively continue through several iterations until a robust architecture is realized.

Approach

Verify Installation

Before attempting any type of reverse and/or forward engineering, it is imperative to confirm that the IBM and Rational software is correctly installed. Verify that VisualAge® for Java™ and Rational Rose J/Enterprise are installed.

Install RoseLink

The RoseLink Plug-In must be installed in order to create connectivity between the products. It is best to install RoseLink after the Rose and VisualAgefor Java products have been installed.

Enable the RoseLink Plug-In in VisualAge

- Launch VisualAgefor Java.
- Click the File menu and choose "Quick Start."
- Choose "Basic" from the Quick Start menu.
- "RoseLink Plugin Toggle" should be visible in the "Basic" list. If it is not, then the RoseLink installation was not successful.
- Click to highlight "RoseLink Plugin Toggle" and click "OK."
Figure 1: If you have installed the plugin correctly, it will be listed in the right column

- A confirmation window will appear.

![Quick Start](image)

Figure 2: A dialog box will let you know when the plugin has started

Create Project(s) Within VisualAge for Java

It is important to note that the integration that occurs between VisualAge and Rose occurs at a VisualAge project level. As such, you must create a project within VisualAge before creating the connection to VisualAge within Rose. This project will hold the packages and classes that will be generated in a forward engineering effort; likewise, it will contain packages and classes to be modeled in a reverse engineering effort.
Enable RoseLink Connection within Rose

- Launch Rose.
- Click the Tools menu.
- Choose "Java."
- Choose "VisualAge for Java Project."
- Rose may prompt you to reset the Java Virtual Machine to "IBM." By choosing "Yes" to the dialog box, it will do this for you automatically. This is actually a Rose model property that can be set manually.

Connect to a VisualAge for Java Project

- The "VisualAge Link Settings" window appears. Choose the VisualAge for Java project to connect to and click "OK."

![Image of RoseLink plugin success message]

Figure 3: Choose the project to which you'd like to connect

- Rational Rose creates a new Project Model property setting called VAJavaProject. This property is set to the project name you selected. In order to view this and other model settings:
  - Click the Tools menu.
  - Choose "Model Properties."
  - Choose "View" or "Edit."
  - Choose the "Java" tab and change "Type" to "Project."

![Image of VisualAge Link Settings dialog box]
Figure 4: On the "Java" tab, set the Type to "Project"

Forward Engineer

In order to successfully forward engineer from Rose to VisualAge for Java, the correct Java frameworks must be loaded and associated with the Rose model. Having the Java frameworks loaded allows Rose to generate the Java code. (See Building Reusable Architecture using Rational Rose Frameworks -- a whitepaper by Khawar Ahmed about the specifics on implementing frameworks with Rose.) In instances where the final implementation is known, it is best to use frameworks, as this allows the designer to model to the specific implementation using the particular constructs associated with that specific language. By default, Rose is packaged with several frameworks available to the designer. However, other frameworks can be created and added to the design efforts.

- To forward engineer from a class diagram, open a diagram in Rose and click to highlight the classes to be generated. Use CTRL-click to highlight multiple classes.
- Right click to view the shortcut menu.
- Choose "Java."
- Choose "Generate Java." Verification messages will appear if the generation was successful; otherwise the Rose log file will detail errors if generation failure occurs.
- Maximize VisualAge for Java and explore to the target project. Notice that packages (if used in Rose) and classes are created under the target project.
Reverse Engineer

Whether this be reverse engineering or the continuation of a roundtrip engineering effort, any changes made to the connected project within VisualAge for Java can be reflected in Rose as a class diagram. Remember that the integration is based upon a specific VisualAge for Java project. In order to reflect alterations within Rose, all changes to packages, classes, and relationships should be made in that project.

- In Rose, click the Tools menu.
- Choose "Java."
- Choose "Reverse Engineer Java."
The "Java Reverse Engineer" window appears showing the VisualAgeconnected project name and package structure.

Highlight the files to be reverse engineered in the upper right-hand window and click "Add."

These files will be added to the lower window.

Click "Select All."

Click "Reverse."

Notice that any new classes, methods, and attributes are reflected in the Rose model.

Assumptions

This effort was based upon the following configuration:
**Windows 2000 Professional**
(http://microsoft.com/windows/default.asp)

**IBM VisualAge for Java 3.5 Enterprise with Service Pack 2 applied**
(http://www-4.ibm.com/software/ad/vajava)

**Rational Rose 2000e**
(http://www.rational.com/products/rose/index.jsp)

**Rose J to VisualAge for Java link**
(http://www.rational.com/support/downloadcenter/upgrades/rose.jsp)

*NOTE:* This article was originally published on Rational Developer Network, the learning and support channel for the Rational customer community. It is part of a three-part series on the integration between Rational and IBM Technologies that includes "Data Model Implementation with Rational Rose and IBM DB2" and "Integrating Rational ClearCase with VisualAge for Java." Rational Developer Network is currently available only to Rational Suite customers. If you are a Rational Suite customer and have not already registered for your free membership, please go to www.rational.net.

---

*For more information on the products or services discussed in this article, please click here and follow the instructions provided. Thank you!*

Copyright Rational Software 2001 | Privacy/Legal Information