

A technical discussion of IBM® Rational® Rapid Developer
Evaluators Guide
June, 2003

Rational® software



IBM® Rational® Rapid Developer Evaluator's Guide

Patrick Luther
lutherp@us.ibm.com

Table of Contents

Who Should Read This Guide?	4
Evaluator Tools and Resources	4
Product Download (Evaluation Software)	4
Knowledge System.....	4
Reference Application.....	4
Online Evaluation (Demonstration Archive).....	5
Quick Start Tutorial.....	5
Self-Paced Training.....	5
Installation and Set-up.....	6
Technical Papers.....	6
Customer Case Studies.....	6
Detailed FAQs.....	6
Evaluating Rational Rapid Developer	7
Approach 1: Online Evaluation.....	7
Approach 2: Quick Start Tutorial.....	8
Topics covered include.....	8
Application Development Challenges	9
Overview.....	9
What to Look for in the "Ideal" Development Platform.....	10
What is Rational Rapid Developer?	11
Overview.....	11
Supported Technologies.....	11
Application Servers.....	11
Databases.....	12
Messaging.....	12
Integration.....	12
Browsers/Appliances.....	12
Security.....	13
Team.....	13
Development.....	13
Rational Rapid Developer Installation/Setup	13
Minimum System Requirements.....	13
Highlighted Product Capabilities	14
Business Modeling	14
Class Modeling.....	14
Business Rules.....	15
Process Architect.....	16
User Interface Modeling	16
Site Architect.....	17
Theme and Style Architects.....	17
Page Architect.....	18
Section 508 Web Accessibility.....	19

Intelligent Controls and the ObjectSpace.....	19
Multi-lingual, Localized Applications.....	19
Database Architect.....	21
Importing Data Models.....	21
Multi-Database Support.....	22
Referential Integrity and Multiple Key Support.....	22
Data Entry.....	22
Sample Data Generation.....	23
Security Architect.....	23
Authentication.....	23
Authorization.....	24
Directory Services Administration.....	24
Messaging.....	24
Legacy Integration.....	26
Components.....	26
Web Services.....	27
Team Development.....	27
Automated Deployment.....	28
Deployment.....	28
FAQs.....	29
What types of application servers does Rapid Developer support?.....	29
What types of database systems are supported?.....	29
What browsers are supported?.....	29
Does Rapid Developer support the J2EE platform and to what extent?.....	30
Does Rapid Developer support Unified Modeling Language (UML)?.....	30
How is this different from a UML modeling tool?.....	30
Can I reuse existing UML models in a new Rapid Developer application?.....	31
What are TechBlades?.....	31
Does Rapid Developer support XML?.....	31
What makes Rapid Developer such a productive development environment?.....	31
How does Rapid Developer provide protections against technology obsolescence?.....	32
Specifically, what code does Rapid Developer generate?.....	33
How does Rapid Developer compare to previous CASE tools?.....	33
How do I modify the application once it is generated?.....	34
Can I override the automatically generated code?.....	34
Do Rapid Developer applications perform?.....	35
What is ObjectSpace?.....	36
Conclusion.....	37

The Evaluator's Guide is appropriate for technical evaluators, developers, architects, press, IT analysts, and anyone who wants to know more about IBM Rational Rapid Developer.

To facilitate the evaluation process, IBM provides a complete set of tools and resources.

Who Should Read This Guide?

The Evaluator's Guide is appropriate for technical evaluators, developers, architects, press, IT analysts, and anyone who wants to know more about IBM Rational Rapid Developer. This guide provides a recommended path to follow for evaluation of the product, as well as a high level product overview including broad feature sets and capabilities of the product followed by a set of frequently asked questions and answers.

Evaluator Tools and Resources

To facilitate the evaluation process, IBM provides a complete set of tools and resources. We have highlighted the key resources in this section.

Product Download (Evaluation Software)

This is a fully functional version of Rational Rapid Developer. Evaluation software is available in the "Try it" section of the Rational Web site <http://www.rational.com/tryit/index.jsp>.

Knowledge System

Rational Rapid Developer Knowledge System is included with the product download and accessed via the Help tab in the product on the main menu bar. The Knowledge system includes:

- *Product Documentation*
- *Context Sensitive Help*
- *API Library*
- *Glossary of Terms*
- *Error Code Descriptions*
- *Product Release Notes*
- *Feature Use Descriptions and Examples*
- *Quick Start Courseware*

Reference Application

The reference application (or "sample" application) serves as a showcase of Rational Rapid Developer functionality as well as an example of common and/or useful application functionality. The sample reference application is an e-commerce exchange system called NeuCommerce. The hypothetical marketplace application is a nationwide system for buyers and sellers of

We have created a library of Viewlets that guides you through the broad feature set and capabilities of Rational Rapid Developer.

The Quick Start tutorial provides evaluators with a step-by-step guide to building and deploying a small e-business application running on a J2EE application server platform.

computer hardware and software. The Web site supports users in the roles of Buyer, Seller, and Administrator. The focus is B2C (business-to-consumer) where Buyers are represented as independent consumers (individual or company) and Sellers as businesses selling products within the marketplace. The reference application ships with the product and can be viewed in design time when launching Rational Rapid Developer by selecting the "open existing application" radio button and selecting the NeuCommerce.ond file.

Online Evaluation (Demonstration Archive)

Since not all potential customers have the time or required technology environment to do a full-scale evaluation, we have created a library of Viewlets that guides you through the broad feature set and capabilities of Rational Rapid Developer. These Viewlets can be accessed by clicking on the following link: <http://www.rational.com/tryit/rrd/seeit.jsp>

Quick Start Tutorial

For evaluators wanting to take Rational Rapid Developer for a test drive, the Quick Start tutorial facilitates the maximum amount of knowledge transfer in the shortest amount of time for first-time users of the product. The Quick Start tutorial provides evaluators with a step-by-step guide to building and deploying a small e-business application running on a J2EE application server platform. The tutorial is designed to be completed in three to five hours and assumes two-plus years of experience with Java, JavaScript, C++, Sybase PowerBuilder, Microsoft Visual Basic or similar.

The Quick Start tutorial courseware can be accessed from the Start button, Programs sub-menu, Rapid Developer selection, "Getting Started with Rapid Developer" tab. The tutorial utilizes two supplementary files that are included in the Quick Start folder in the Rapid Developer directory at <X>\Rapid Developer\Samples\QuickStart.

Self-Paced Training

The product Quick Start contains the courseware for the two-day, instructor led "Introduction to Rapid Developer" class. The version of the courseware in the Quick Start has been designed to be completed as self-paced training without the need for an instructor.

Installation and Set-up

A full-scale evaluation of Rational Rapid Developer may require some configuration to your current computing environment. The Installation and Set-Up Guide provides detailed information on how to configure your environment based on your desired outcome, e.g. COM+ configuration is different than J2EE.

Please refer to the Knowledge System under the "Using Rapid Developer" tree navigation on the left, sub-menu choice "Installation and Set-up". The Quick Start also contains information on how to configure your environment for completion of the Quick Start.

IBM Rational publishes a series of technical papers on Rational Rapid Developer.

Technical Papers

IBM Rational publishes a series of technical papers on Rational Rapid Developer. We encourage you to reference these during your evaluation. Here is a partial list of topics:

- *Automated construction*
- *Scalability and performance*
- *Globalizing Web-based systems*
- *Legacy integration*
- *Components and Web services*
- *Architected RAD and enterprise architecture*

You can find these on the Web at:

<http://www.rational.com/products/rapiddeveloper/index.jsp>

IBM Rational publishes a series of customer usage scenarios (PDF and video formats) describing the use of Rational Rapid Developer.

Customer Case Studies

IBM Rational publishes a series of customer usage scenarios (PDF and video formats) describing the use of Rational Rapid Developer. We encourage you to review these during your evaluation.

You can find these at:

<http://www.rational.com/products/rapiddeveloper/index.jsp>

Detailed FAQs

We have included answers to some of the most frequently asked questions in the pre-sales phase in the FAQs section of this document.

IBM Rational offers two approaches to evaluating Rational Rapid Developer.

Evaluating Rational Rapid Developer

To provide an evaluation process that meets your needs, IBM Rational offers two approaches to evaluating Rational Rapid Developer. To determine which path is appropriate for you, refer to the table below.

Table 1: Evaluation Choices

Item	Approach 1 Viewlets	Approach 2 Quick Start
Audience/ Suitability	Appropriate for evaluators who want to quickly understand the product capabilities, non-technical users, those with no access to deployment technologies, or those more comfortable viewing rather than doing.	Appropriate for evaluators who want to test-drive the product. In this Quick Start tutorial, you will develop, test and deploy a small e-business application from scratch, on a J2EE server platform. Two or more years of Microsoft Windows development and familiarity with either Java, JavaScript, C++, PowerBuilder or Visual Basic is required.
Estimated Time Commitment	Various depending on number of Viewlets watched	Approximately 3 to 5 hours.
How to Proceed	Click on link below. Select desired Viewlet and watch demo. http://www.rational.com/tryit/rrd/seeit.jsp	Upon successful installation of Product, go to the Start button, Programs, Rapid Developer selection, and click on "Knowledge System". From here, select the hyper-link to the Quick Start Tutorial.

Approach 1: Online Evaluation

We recommend this approach for evaluators who need a quick understanding of the product capabilities in the shortest amount of time. The Product Tour is the best general introduction to Rational Rapid Developer. Click on the following link <http://www.rational.com/tryit/rrd/seeit.jsp> to be re-directed to the page on the IBM website where the Viewlets can be accessed.

This Quick Start is designed to be completed in three to five hours.

Approach 2: Quick Start Tutorial

This approach is recommended for a test drive of Rapid Developer at your own pace. The Quick Start tutorial provides evaluators with a step-by-step guide to building and deploying a small e-business application running on a J2EE application server platform. This Quick Start is designed to be completed in three to five hours and assumes two years of development experience and familiarity with Java, JavaScript, C++, Visual Basic or PowerBuilder.

Topics covered include:

- *Rational Rapid Developer navigation (RAD environment)*
- *The major components and architecture of a Rational Rapid Developer project*
- *Rational Rapid Developer generated code and directories*
- *Overview of Rapid Developer development methodology*
- *Creating a Class Model using:*
 - *Class import from Rational Rose (UML)*
 - *Attributes*
 - *Business rules*
- *Steps in e-business development using Rational Rapid Developer*
- *Creating a Site Model*
- *Creating several types of dynamic, data-connected Web pages*
- *Constructing the business logic, and creating various classes and methods*
- *Demonstrating automatic parameter-passing*
- *Testing your application using a J2EE application server*

The tutorial utilizes two supplementary files called QuickStart.xmi and uml.dtd that are included in the Quick Start folder in the Rational Rapid Developer product at <X>:\Rapid Developer\Samples\QuickStart.

Application Development Challenges

Overview

Global organizations face mounting challenges in developing and delivering applications that will meet their needs today, and more importantly, their needs of tomorrow. These challenges include:

- *Doing more with less – increased application development requirements with reduced budgets and/or staff*
- *Taking legacy systems forward and enabling them for XML, Web services, Web and wireless*
- *Harnessing the business domain knowledge of existing developers while enabling them to create complex, n-tier applications*
- *Leveraging emerging technologies such as XML, Web services and wireless*
- *Meeting faster business and technology cycle times to deliver new applications and upgrades*
- *Protecting IT investments and people from becoming technologically obsolete*
- *Improving the quality, scalability and reliability of delivered systems*
- *Creating collaboration and improving synergies between business units and IT developers*
- *Solving the shortage of skilled developers, which hampers an organization's ability to find and retain staff*
- *Reducing total cost of ownership*
- *Creating and maintaining multi-lingual applications*

While most vendors will address one or several of these challenges in individual products, the vendors addressing all or even most of these issues in a single product is limited or nonexistent. These issues and challenges are compounded when companies are forced to buy multiple non-integrated "point" solutions to address these problems.

What to Look for in the "Ideal" Development Platform

What these demands/challenges have made perfectly clear is that the solution does not come from a "traditional" development tool or set of tools. Below is a table that summarizes the key features and characteristics of the ideal development platform.

Table 2. Features of the Ideal Development Platform

Feature/Characteristic	Description
1. Platform/technology neutral	Tool should support deployment to a variety of technologies across all tiers; multiple databases, legacy systems and user interfaces and allow for easy migration from one version of a technology to another (e.g. WebSphere 4.0 to 5.0.)
2. Integrated and end-to-end development environment	Applications have multiple tiers, so the ideal tool should also address multiple tiers. Integrated means developers don't need multiple tools for multiple tiers as well as the headaches that go along with meshing inputs and outputs from cobbled together point solutions.
3. Support for full development life cycle	Tool should allow for consistency across the development life cycle from requirements gathering through testing. Developers need tools that allow for traceability of requirements and use cases either natively or through tight integration with other products, as well as the ability to work in a team development environment.
4. Rapid development through visual modeling and automated construction	Hand coding takes too long. Hand coding n-tier applications take way too long. The only way to address speed of development is to get away from hand coding. This is accomplished by automatically generating code from models. However, the code generated needs to go beyond stubs and shells (which can be helpful, but are a long way from a completed application). Instead, the models need to be sufficiently robust and complete so that they generate fully executable applications.
5. Easy to use	With Java's success, there is a general shortage of Java developers today, especially J2EE developers capable of developing n-tier systems. The ideal tool will leverage business and domain expertise, traditional application development and visual modeling skills, and focus limited Java skills on high value custom business logic and deployment issues. The result is high productivity from non-Java resources.

Table 2. Features of the Ideal Development Platform

Feature/Characteristic	Description
6. Flexible	Business changes so fast that the ability to modify quickly, whether it is business logic, look and feel, or deployment technologies, is a crucial feature.
7. High-end, n-tier enterprise applications	IT organizations want tools that can create simple, departmental applications as well as standards-based, scalable, enterprise-wide applications that integrate legacy data and support the various channels of Web (HTML), wireless (WML), messaging (XML) and Web services (SOAP).

What is Rational Rapid Developer?

Overview

Rational Rapid Developer is a model-driven, architected rapid application development (ARAD) environment. It provides a single, integrated development environment that combines model-driven development, RAD techniques and automated construction to rapidly deliver well architected, fully executable, n-tier business applications. By insulating developers from most of the traditional complexities of n-tier development and the underlying technology platforms, Rational Rapid Developer enables a broad class of developers to participate in the creation of enterprise-class applications.

Rational Rapid Developer enables a broad class of developers to participate in the creation of enterprise-class applications.

Supported Technologies

The Rapid Developer environment is technology-neutral and supports optimized deployment on a range of popular technologies. These technologies include:

Application Servers

J2EE

- *IBM WebSphere 3.5, 4.0, 5.0*
- *BEA WebLogic 5.1, 6.1, 7.0*
- *Oracle 9iAS*
- *Apache Tomcat 4.0, 4.1*

Microsoft MTS

Databases

- *IBM DB2 UDB v6.0, 7.2, 8.0*
- *IBM DB2 eServer zSeries (S/390) v6, 7, 8*
- *IBM DB2 iSeries (AS/400) V4R5, V5R1*
- *IBM IMS*
- *IBM VSAM*
- *Microsoft Access*
- *Microsoft SQL Server 7.x, 2000*
- *Oracle 7.3, 8/8i, 9i*
- *Sybase 11*
- *60-plus data sources via iWay Adapters*

Messaging

- *XML*
- *EDI via Third-Party Platforms*
- *IBM WebSphere MQ (MQ Series)*
- *Java Messaging Service (JMS)*
- *MSMQ*
- *TCP/IP*
- *TIBCO*

Integration

API

- *EJB, CORBA, COM*
- *Mainframe: CICS, MVS*

Data Connectors

- *Mainframe: IMS, CICS/VSAM*
- *60-plus data sources via iWay Adapters*
- *All supported relational databases*

Web Services

Browsers/Appliances

HTML

- *Microsoft IE 4.0+*
- *Netscape Navigator 3.0+*
- *Other compatible*

WML/WAP

Security

- *Digital Certificates*
- *LDAP*
- *Role-Based Access Control*
- *SSL*

Team Development

- *IBM Rational ClearCase*
- *IBM Rational Unified Process: RUP® Plug-in for Rational Rapid Developer*
- *Merant PVCS*
- *Microsoft Visual SourceSafe*
- *Products supporting SCC API*

Rapid Developer Installation/Setup

Minimum System Requirements

Hardware

A high-performance IBM-compatible computer with the following minimum configuration:

- *Pentium III-based microprocessor (minimum 300 MHz)*
- *256 MB minimum (768 MB for IBM WebSphere)*
- *500 MB disk space (recommended)*
- *A high-resolution display monitor (minimum 1024 x 768)*

Operating Systems

Rapid Developer will run on the following operating systems:

- *Windows NT 4.0 (Workstation or Server) – Service Pack 5*
- *Windows XP*
- *Windows 2000 – Service Pack 1*

Note: Applications constructed from Rational Rapid Developer can be deployed on any operating environment (operating system / hardware) supported by the underlying deployment platform (ex: IBM WebSphere running on an eServer zSeries machine, or BEA WebLogic running on a UNIX box).

Highlighted Product Capabilities

Business Modeling

Class Modeling

The Class Architect is Rational Rapid Developer's object-oriented, visual class modeling tool. It allows you to focus on analyzing the business domain and translating the business domain into an appropriate object design. The object-oriented focus produces smaller systems through the reuse of common components and produces systems that are more resilient to change and better able to evolve over time. The Class Architect's design features directly address the inherent complexities of software by helping you make intelligent decisions.

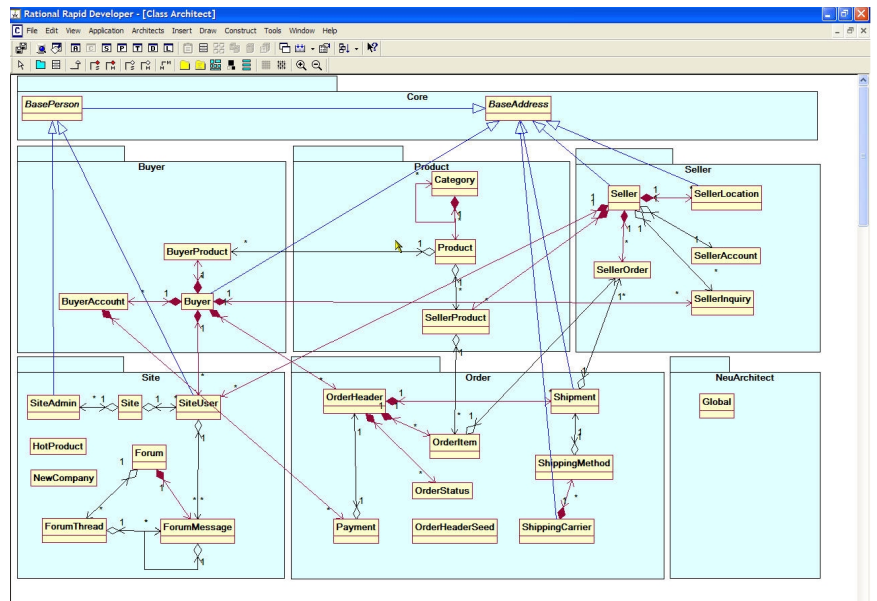


Figure 1. Class Architect

Class models support an unlimited number of classes: abstract, non-persistent classes, and persistent classes.

Using Class Architect, a business expert's understanding and requirements are directly captured as class models. Class models support an unlimited number of classes: abstract, non-persistent classes, and persistent classes. You define class attributes and methods, define inheritance relationships, and capture associations such as multiplicity, navigability, and composition or ownership relationships. The resulting class models form the superstructure of the application (e.g., data, middle-tier and client user interface) and allow developers to leverage this information directly and efficiently throughout the

integrated visual tools in Rational Rapid Developer. Whether you are designing a Web page, extending your object with custom business logic, or designing XML messages, modeled class information is available to rapidly and accurately build your application.

Class Architect uses UML-compatible diagrams so you don't have to learn a new modeling notation. UML notation for packages, classes and associations is supported. You can quickly and intuitively display the amount of information you want to see for each class. Rational Rapid Developer allows direct import of class diagrams, leveraging your existing investment in tools and legacy systems. Imports utilize the open, industry-standard XML Metadata Interchange (XMI) interface. Rational Rapid Developer also provides a feature for synchronizing with IBM Rational Rose and IBM Rational XDE models. This feature provides a facility for direct import of Rational Rose or Rational XDE class models into Rational Rapid Developer without the need for XMI translation. It allows you to keep the Rational Rapid Developer model in sync with the Rational Rose or Rational XDE model while also supporting the ability to filter any elements from the synchronization. Rational Rapid Developer extends these imported models into complete application models, adding methods, business rules, pages, messaging and other attributes. Rational Rapid Developer is the fastest way to go from UML models to complete n-tier executable applications.

The Business Rules modeling in Rational Rapid Developer provides the ability to attach language-based expressions to an attribute so you can extend the semantics of an attribute at the object level.

Business Rules

The Business Rules modeling in Rational Rapid Developer provides the ability to attach language-based expressions to an attribute so you can extend the semantics of an attribute at the object level. Business rules are simply business logic and should be maintained by business analysts. Typical business rules that can be quickly defined and deployed are rules for initial attribute value, derivation value, and validation rules.

- *Initial value rules are used to provide initial values for attributes.*
- *Derivation rules are used to calculate the value of an attribute, based on the values of other attributes in the same or related classes.*
- *Validation rules are used to validate the value of the attribute. It may validate the value against constant values (e.g., a range check), or against other attributes (e.g., "order total" must not exceed the maximum allowed for the customer).*

Process Architect

The visual modeling features incorporated in Process Architect help an application designer or analyst to accurately model the business processes that must be embodied and supported by the application.

You use the Process Architect to describe a process or sequence of activities and related states. The Process Architect captures state and activity. Modeling elements include states, methods, outcomes, branches, merges, and decisions.

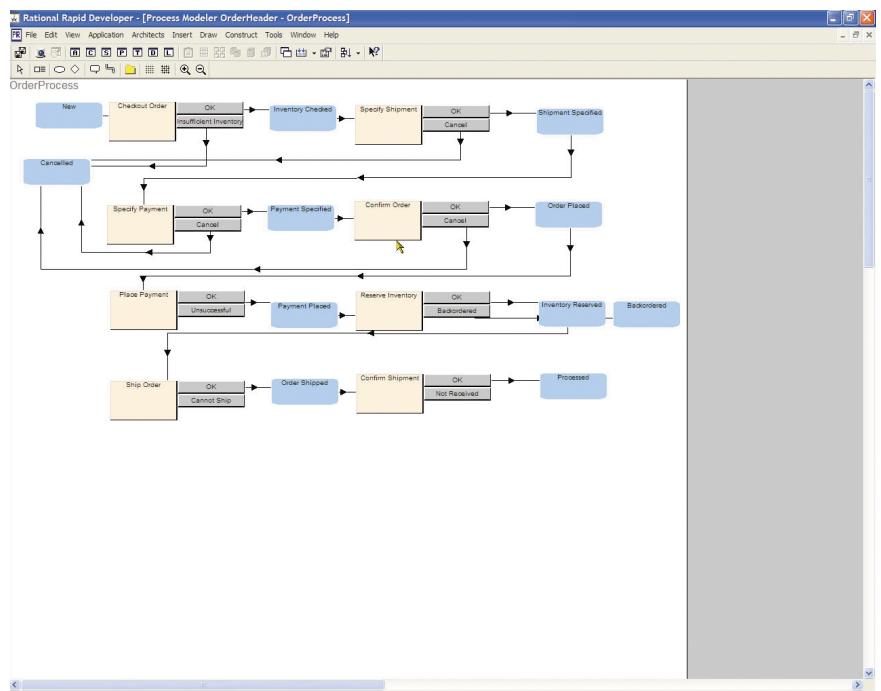


Figure 2. Process Architect

User Interface Modeling

Rational Rapid Developer provides a comprehensive system for user interface modeling. It uses the Site Architect to model user navigation, the Theme Architect for consistent look-and-feel, and the Page Architect to visually design the user interface. The most unique component is the transaction model embedded in each page that links it to business objects and data. This enables the page to provide full application interaction.

The style repository within Rational Rapid Developer provides the ability to create graphical style guidelines and templates defining color, font and other appearance attributes for the pages in the site and each type of control. This permits the lead graphic designer to specify a unique look-and-feel for the entire or part of the Web site under development. These style sheets can also be designed for performance with an emphasis on minimizing page size and improving page load efficiency. All developers on the project can then use the style sheets to create pages optimized for small page size and quick loading.

Page Architect

The WYSIWYG Web page authoring tool, Page Architect, contains a rich set of graphical controls to support state-of-the-art user interface design. In addition, documents created with Page Architect are automatically connected to the middle-tier through the code construction facility. This powerful feature eliminates the need to write presentation logic for building data-connected Web pages. Page Architect supports both HTML-based browsers and WML/WAP-based devices. With Rational Rapid Developer, it is also easy to integrate existing visual components such as Java applets, custom HTML controls, and ActiveX components. These are simply placed on the document pages and can be easily customized to meet specific requirements.

Documents created with Page Architect are automatically connected to the middle-tier through the code construction facility. This powerful feature eliminates the need to write presentation logic for building data-connected Web pages.

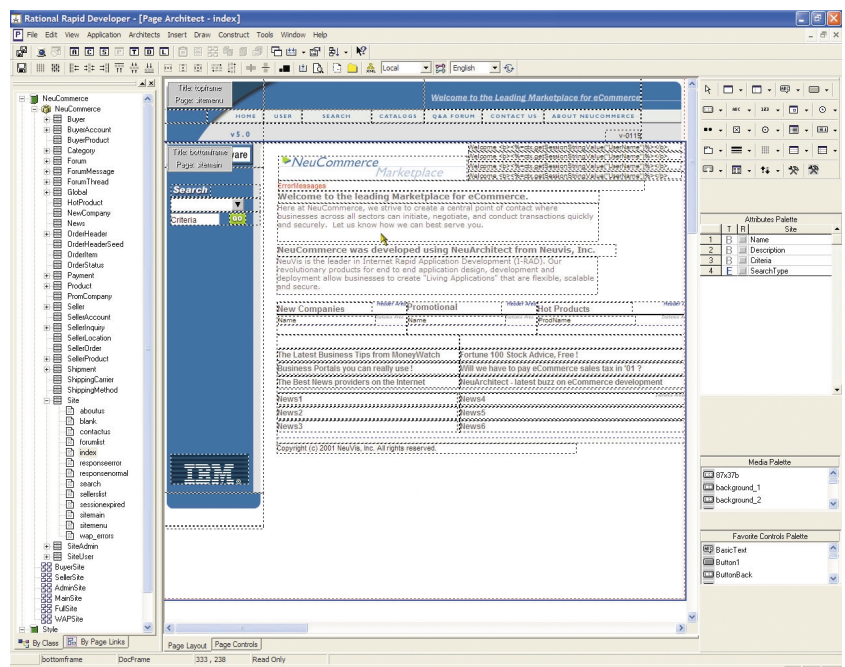


Figure 4. Page Architect

The ObjectSpace for a Web page determines the definition of business object instances created on the Web server or application server at run-time.

For more information, see *IBM Rational Rapid Developer, Globalization and Creating Multi-Locale Applications* technical paper.

Section 508 Web Accessibility

Rational Rapid Developer provides support for Section 508 Web accessibility requirements. In 1998, the United States Congress amended the Rehabilitation Act to require U.S. Federal (central) agencies to make their electronic and information technology accessible to people with disabilities. Section 508 was enacted to eliminate barriers in information technology, to make available new opportunities for people with disabilities, and to encourage development of technologies that will help achieve these goals. Rational Rapid Developer has RAD support for these requirements. Support includes items such as; ALT tag or help field text support, column and row header tags, cell headers, frame titles, and more.

Intelligent Controls and the ObjectSpace

ObjectSpace refers to the Rational Rapid Developer-created "workspace objects" modeled and produced by the constructed code. The ObjectSpace handles data mapping and transaction processing between browser, business objects, and database. The ObjectSpace for a Web page determines the definition of business object instances created on the Web server or application server at run-time. The developer has complete control as to what entities, methods, and attributes are projected into the ObjectSpace enabling optimized business transactions. These optimized business object instances serve the Web pages and handle presentation and persistence logic, doing things like populating the page with browser data (either data being sent to the browser or being sent by the browser via form posting), and reading or writing data to and from the database. This feature provides tight integration between graphical controls and the underlying data. For example, writing code to display data is unnecessary because this logic is done automatically by Rational Rapid Developer's ObjectSpace processing.

Multi-lingual, Localized Applications

Rational Rapid Developer provides built-in capabilities for rapidly creating, deploying, and maintaining multi-lingual, multi-locale applications. Just as with all other features of Rational Rapid Developer, the ability to create applications that support multiple language, formats, and locale specifics is a rapid and easy process. The Rational Rapid Developer localization model abstracts away the many pains in creating these applications, giving you an unparalleled competitive advantage and cost savings in creating and maintaining international/localized applications.

For more information, see *IBM Rational Rapid Developer, Globalization and Creating Multi-Locale Applications* technical paper.

Table 3. Rational Rapid Developer Localization Model Solution

Table 3. Rational Rapid Developer Localization Model Solution	
Problem	Solution
Internationalization and localization problem space and 'how-to' is complex and variable.	We have completely researched the technical considerations involved with internationalization and have packaged the solution in an easy-to-use, flexible RAD environment.
Difficult and costly to train a entire team on the complexities of internationalization (i18n).	The localization model insulates the application developer from the complexity of i18n without sacrificing flexibility. The application developer can focus on the task of building the application without worrying about the complexity of localization.
The process of translation can be cumbersome, costly, and inefficient.	Rational Rapid Developer provides an integrated system to send and receive translations from translators. The information is sent in industry standard formats (TMX, XLIFF), Excel spreadsheet format, or CVS files. Translated strings are automatically integrated back into the application model.
It is difficult to create and manage text strings across languages.	Rational Rapid Developer insulates the developer from this complexity. The developer does not have to work with resource bundles. Resource bundles are automatically generated and deployed.
It is difficult to determine the effect of translation on page design and layout	In traditional development environments, you must do this at runtime. With the visualization provided by the Rational Rapid Developer Page Architect, a developer or translator can immediately see the effects of translation on the page layout. We have added another dimension to WYSIWYG, that is multiple language support.
Images are hard to manage across languages.	Rational Rapid Developer automatically manages images across multiple languages.
It's very difficult to localize an application once it has been written.	There is no need to decide to localize up front. The Rational Rapid Developer localization model facilitates the migration of non-localized, Rational Rapid Developer applications to a multi-language applications.
Multiple versions of pages, each in a different language, are very difficult to maintain	Rational Rapid Developer allows you to design one page that displays in different languages (determined by the operating system language settings on the user's computer), greatly reducing maintenance.

Database Architect

You define class persistence using database modeling. Rational Rapid Developer's database modeling enables you to map the logical model to a physical model, while providing the necessary abstraction that gives you the ability to quickly change your deployment database. Rational Rapid Developer's database modeling capabilities allow you to:

- *Construct physical database persistence layers as defined by the class model.*
- *Resolve differences between the modeled and an existing constructed database layer, and then alter an existing database to reflect new models.*
- *View all DDL so that you can create and drop statements before constructing persistence.*
- *Set database-specific optimization parameters.*

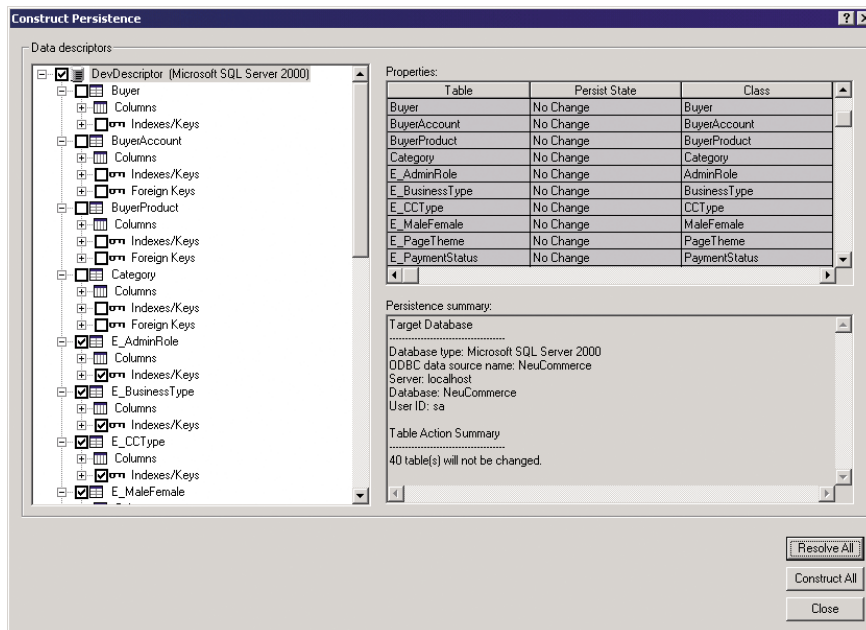


Figure 5. Constructing Persistence

Importing Data Models

Rational Rapid Developer protects legacy investments by supporting reverse engineering of existing databases. With Rational Rapid Developer's database import wizards, you can take existing database schemas and create classes from them for use in your own application. You can accomplish this using a

direct connection to the database, or using an XML-based catalog import. Class relationships can also be created, based on foreign key relationships in the database.

Multi-Database Support

Rational Rapid Developer provides the ability to specify the data source for each class being modeled, facilitating integration with existing legacy systems. The data source can be any database supported by Rational Rapid Developer. For example, Product and Category classes may be persisted in a IBM DB2 database, while Customer and Account classes can be persisted in an Oracle database. Rational Rapid Developer ensures that the persistence interface is directed to the appropriate data source.

Referential Integrity and Multiple Key Support

Referential integrity describes the mandatory existence of parent-row primary key values for dependent-row foreign key relationships. Rational Rapid Developer enforces referential integrity if it is defined in the business model. Rational Rapid Developer provides the capability to create relationships between entities based on multiple keys. This allows modeling of complex relationships in which a parent entity's key becomes part of its child's composite key, and so on. Rational Rapid Developer also:

- *Supports cascaded relationships and generate appropriate triggers if database requires and generates appropriate DDL.*
- *Maintains relationships and supports complex primary keys and custom matching relationships.*
- *Provides built-in support for relationships via surrogate keys.*
- *Supports domains for reducing joins.*
- *Abstracts JDBC complexities from the user.*

Data Entry

The Rational Rapid Developer Data Entry feature facilitates testing by allowing you to directly enter data without using external tools or scripts for data creation. You can use this feature in conjunction with the Sample Data Generation feature to quickly set up any amount of test data.

Rational Rapid Developer provides the capability to create relationships between entities based on multiple keys. This allows modeling of complex relationships in which a parent entity's key becomes part of its child's composite key.

A common gateway allows Rational Rapid Developer-constructed applications to work with the most popular Directory Service technologies at runtime.

Sample Data Generation

Finding adequate sample data is one of the biggest problems in the development and testing of sites. You can use Rational Rapid Developer's Sample Data Generation feature throughout the development cycle for prototyping, development, testing, and training. From the developer's point of view, this feature facilitates prototyping by eliminating data issues. From the tester's point of view, there is always the need to generate test data for sites having varying volumes of data. Rational Rapid Developer eliminates the need to find additional tools to generate data or the need to write a program to generate the data. For applications with a complicated object model, you can use Sample Data Generation to instantaneously test the site.

Security Architect

Applications leveraging Rational Rapid Developer's security framework are not constrained to any particular vendor of Directory Services or technology. A common gateway allows Rational Rapid Developer-constructed applications to work with the most popular Directory Service technologies at runtime. These include LDAP server products offered by Sun, Netscape, Microsoft's Active Directory, or other relational database directory services. The Rational Rapid Developer security framework supports simple password, X.509 digital certificates, smart cards, biometrics, authentication timeout techniques, and additional security standards. Multiple Security Provider instances allow an application to connect to multiple types of Directory Services. Rational Rapid Developer supports data and role-based security.

Authentication

Each generated Web site creates and maintains a user profile that includes information such as name, title, address and phone number. Profiles can also include global information like departments, groups, or divisions. The Rational Rapid Developer platform provides role-based management of access rights and their presentation to the Rational Rapid Developer developer. A role is a set of access privileges assigned to a user when their profile is created. Profiles and roles are maintained through the directory manager's administration facility.

Authentication is performed at runtime. Each resource request generates a search against the Rational Rapid Developer Security Provider to verify that the user has current authentication privileges.

Authorization

Rational Rapid Developer application developers can personalize content down to the control level within a document based on the user's identity. Users see only the resources that they are entitled to use while restricted functions remain hidden. If the user has insufficient privileges, the Rational Rapid Developer developer can set the action to automatically redirect the user to another resource.

Encryption is performed using a Secure Sockets Layer (SSL) connection to maintain secure data transfer during authentication and authorization operations. Data such as username and passwords are protected during transmission by an infrastructure that keeps unauthorized parties from reading or modifying data. URL parameter and XML message encryption is also supported.

Directory Services Administration

The Rational Rapid Developer security framework provides the ability to change schema mapping or directory service settings without regenerating Rational Rapid Developer documents or objects or rebuilding run-time components. This is especially important during the development, testing, and deployment phases of application building.

Messaging

Rational Rapid Developer supports the ability to integrate with other enterprise or partner systems in a variety of ways including leveraging messaging systems and technologies. As with other features in Rational Rapid Developer, you can rapidly model your inbound and outbound messages, both XML or custom, to support your specific business needs. The developer will focus on defining what entities, attributes, and methods are required to support the specific transaction (such as processing an inbound customer order), what business methods to be automatically invoked, and mapping the message to the appropriate ObjectSpace or transaction. Rational Rapid Developer will handle the rest.

You can rapidly model your inbound and outbound messages, both XML or custom, to support your specific business needs.

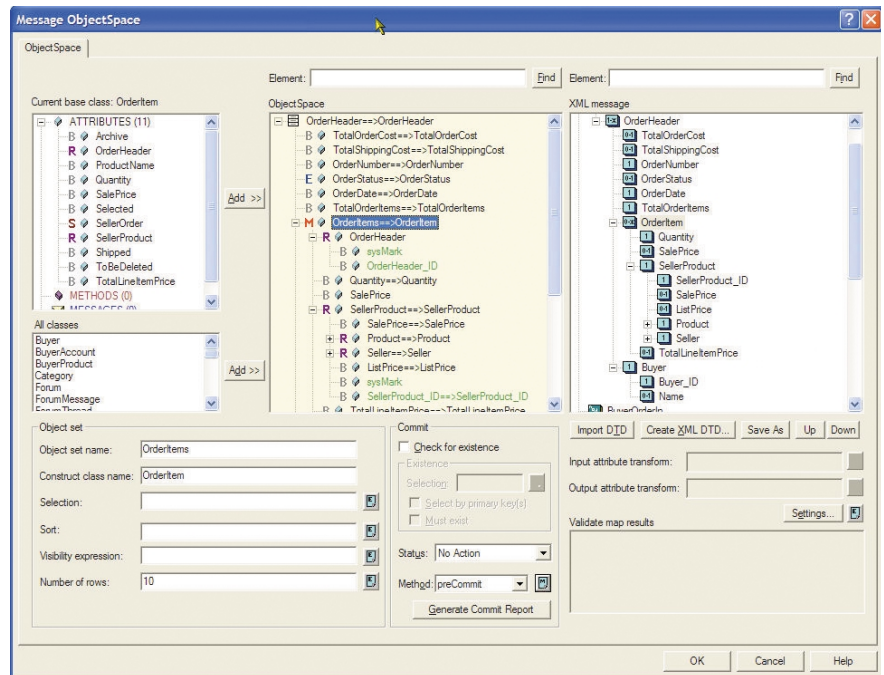


Figure 6. Message Modeling

Rational Rapid Developer's messaging framework provides a sophisticated multi-threaded message broker (or you can utilize J2EE Message Driven Beans) to process inbound messages and automatically route the message to the correct component. Inbound message queues can be configured for any number of concurrently executing threads, depending on machine resource limits. Outbound messages are sent directly through the messaging system to their destination. Users can define as many different input queues as needed, limited only by the hardware resources. A variety of messaging systems are supported, including:

- *Native support for IBM WebSphere MQ (MQ Series), JMS providers, TIBCO, and MSMQ*
- *XML messages and DTDs*
- *Plain text messages*
- *Synchronous and asynchronous messaging*
- *Complex transformations*
- *Automated construction of parser and formatters based on the XML to the DTD mapping*

Legacy Integration

Traditional legacy systems have evolved with limited thought for future integration requirements. Many of these older systems were developed or purchased for a specific department within the company or designed to solve a narrow set of problems. Most legacy systems are unable to interface beyond their own environment. In this era of diverse technology solutions, any integration issue quickly becomes an integration problem. Rational Rapid Developer addresses these problems.

Rational Rapid Developer provides the ability to interface with existing legacy applications on the IBM mainframe platform – both databases and enterprise application services. The Rational Rapid Developer mainframe integration adapters enable you to access IMS or VSAM data without having to understand the mainframe details or technologies. They are treated as yet another data source for your application developers to access. Likewise, you can access and harness both visual and non-visual CICS transactions from Rational Rapid Developer in the same fashion.

Rational Rapid Developer also provides support for many commercial mainframe solutions in order to rapidly leverage “green screen” applications in your new or composite application. So the problem of getting information buried deep in green screen transactions and integrating this information with the rest of your application or enterprise, gets solved easily with Rational Rapid Developer.

For more information, see *IBM Rational Rapid Developer, A Guide to Legacy Integration* technical paper.

For more information, see *IBM Rational Rapid Developer, A Guide to Legacy Integration* technical paper.

Components

With Rational Rapid Developer, you can model components that enable support of a services oriented architecture. These components leverage your information model and can be optimized for the services they support. For J2EE, these components translate to Session beans at construction time. These can be stateless or stateful, or even construction to non-J2EE, Java packages. These components have user-defined interfaces and can be automatically constructed and deployed. These constructed components can be consumed by any J2EE application, and are ideal for reuse across the enterprise.

Web Services

Rational Rapid Developer supports the discovery and rapid use of existing Web services. In addition, Rational Rapid Developer-modeled components can be exposed as Web services with a click of the mouse. These Web services can also be automatically registered to a UDDI registry of your choice. Using Web services in Rational Rapid Developer is rapid and easy. The developer is insulated from SOAP protocols and underlying APIs.

For more information, see *IBM Rational Rapid Developer, Components & Web Services technical paper*.

For more information, see *IBM Rational Rapid Developer, Components & Web Services technical paper*.

Figure 7. Using Web services

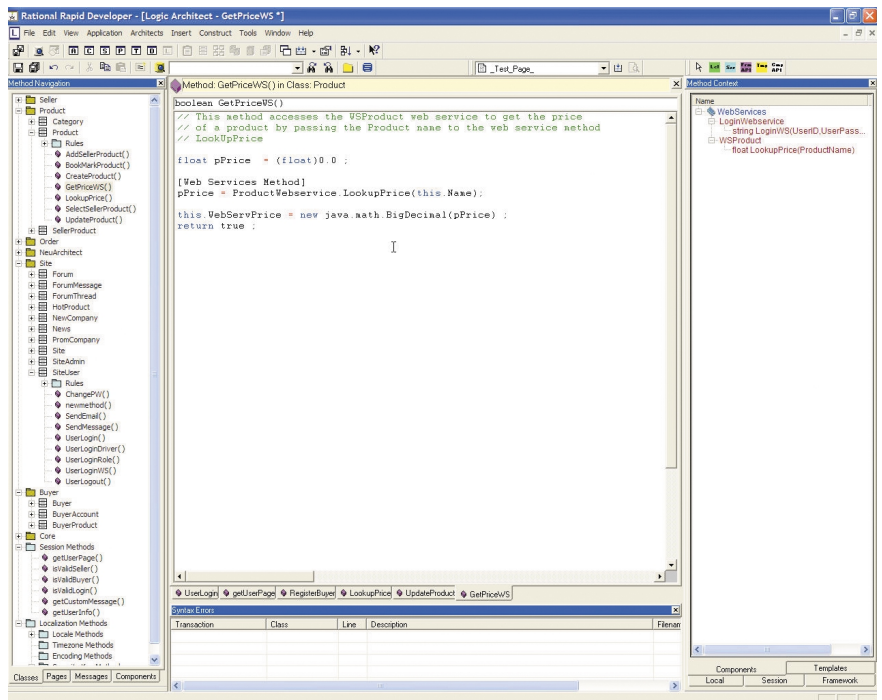


Figure 7. Using Web services

Team Development

Rational Rapid Developer supports a true multi-user development environment. Team Development support enables project team members to perform version control activities such as check-in/out and update functions on project components. The team development environment supports IBM Rational ClearCase as well as other commercial version control systems that utilize the Microsoft SSC API (Microsoft Visual Source Safe, Merant PVCS, MKS Source

Integrity, etc.). Integrated features include the ability to: Add and Remove from Source Control, Get Latest Version, Check In, Check Out, Undo Check Out, Disconnect from Source Control, Open Source Control Project, and Run Source Control. Support for automated backups is also available.

Automated Deployment

Deployment of a business application in an iterative development process can be a timely and frustrating activity. Rational Rapid Developer minimizes the burden of deployment so that the developer can concentrate on the task at hand, namely application development. When developing on a local machine, the developer simply presses the construct button. Your code is generated, compiled, and then locally deployed automatically no matter what deployment technologies you have selected. Rational Rapid Developer makes it easy.

For remote deployment to staging and production environments, Rational Rapid Developer creates the appropriate deployment artifacts, easing your remote deployment tasks. This is done using the Partition Architect. The Partition Architect enables you to specify any number of deployment environments and technologies and will optimally construct the deployable artifacts for these environments. Functional and physical partition capabilities are enabled.

For more information, see *IBM Rational Rapid Developer, Automated Construction* technical paper.

For more information, see *IBM Rational Rapid Developer, Automated Construction* technical paper.

Deployment

No truly end-to-end development environment would be complete without a large menu of deployment choices. One of the fundamental concepts of Rapid Developer is that the application design is "insulated" from the deployment technologies. This layer of abstraction enables Rapid Developer to support many of the industry-leading technologies today, and provides the ability to readily add support for new technologies in the future, (e.g. voice recognition, various handheld devices, etc). In addition, the separation of the application logic from the deployment technologies allows Rapid Developer to migrate applications from one technology to the next with the click of a mouse, literally. The application is simply optimized and reconstructed for the newest target technology, and the old code is discarded.

FAQs

Rational Rapid Developer is leading-edge application development technology. Architected rapid application development (ARAD) – model-driven development coupled with automated, n-tier construction, supporting cross-platform, multi-channel systems. This is sometimes viewed as a future, idealistic objective – but IBM Rational has it today. So we sometimes get skeptical questions. Here we have provided the most frequently asked questions (FAQs) and accompanying responses.

What types of application servers does Rapid Developer support?

Rapid Developer supports J2EE application servers as well as Microsoft DNA.

For Java environments, the Rational platform supports all J2EE application servers. Rapid Developer provides automated deployment to IBM WebSphere, BEA WebLogic, Oracle 9iAS, Apache Tomcat as well as a generic J2EE construction choice. Rapid Developer-generated applications will manually deploy to all other J2EE application servers.

For the Microsoft environment, Rapid Developer supports COM objects on Windows NT and Windows 2000. This includes Internet Information Server (IIS), Microsoft Transaction Server(MTS), and MSMQ.

What types of database systems are supported?

The Rational platform supports popular relational databases, including IBM DB2, Microsoft SQL Server, Microsoft Access, Oracle and Sybase. The Rational platform directly supports IBM IMS, IBM VSAM (CICS and MVS application programs) and via integration with iWay Software's Adapters, provides access to more than 60 data sources.

What browsers are supported?

The Rational platform supports popular browsers, including versions of Microsoft Internet Explorer and Netscape Navigator. The HTML template languages supported include Microsoft's Active Server Pages and Java Server Pages. It also supports Java Applets, Shockwave and Flash.

Does Rapid Developer support the J2EE platform and to what extent?

Yes. Rapid Developer supports J2EE and allows users to automatically generate Enterprise Java Beans as part of the constructed application. Rapid Developer also supports the leading EJB standard application servers, including IBM WebSphere, BEA WebLogic, and Oracle 9iAS.

Does Rapid Developer support Unified Modeling Language (UML)?

Yes. Rapid Developer includes support for UML class models. Importantly, Rapid Developer includes additional models that extend UML design in order to facilitate automated code construction of applications envisioned in the models. In addition to the UML class models, the Rapid Developer Modeling System also includes models for database, process, security and rules; user interface models for theme, style, page, site interaction (linking business objects to data); and integration models for external (B2B) and internal (EAI) systems using database, XML, API and Web services.

How is this different from a UML modeling tool?

There are a couple of key differences between the two categories of products. The first is that UML tools provide a broader set of design models and have been traditionally more focused on modeling. The subsequent creation of an application based on the models is not necessarily the purpose of the tool itself, but rather more of an IDE function although newer versions of UML tools have incorporated code generation capabilities. On the other hand, Rapid Developer is intended entirely for the single purpose of creating an application from models. Therefore, every model used in Rapid Developer has some impact on the generated code, which is not the case with UML tools. Another difference between the two categories of tools is in the models that they employ. Both products use class models, however, because Rapid Developer generates fully executable applications from models, it needs to go beyond UML to model details that UML does not provide for but are needed in an application. Those models include pages, site navigation, styles and themes, messaging, databases, processes, integration and deployment. The end result is the complete modeling of an application followed by automated code construction of the complete, executable application, not just a subset of the application.

Can I reuse existing UML models in a new Rapid Developer application?

Yes. You can import existing UML class models into Rapid Developer as business objects using XML Metadata Interchange (XMI). Rapid Developer also provides import and iterative synchronization with Rational Rose and XDE class models leveraging Rational's Extensibility APIs. Rapid Developer extends these imported models into complete application models, adding methods, business rules, pages, messaging and other attributes. Rapid Developer provides the fastest path from UML models to complete n-tier executable applications.

What are TechBlades?

Rapid Developer TechBlades™ are code construction templates that automatically optimize Rapid Developer-generated code for specific deployment technologies, including leading application servers, databases, messaging transports, browsers, security infrastructure and Web services. By employing these interchangeable TechBlades, Rapid Developer lets developers select deployment technologies with the click of the mouse to create applications optimized for each technology.

Importantly, these TechBlades protect Rapid Developer-based applications from technology obsolescence, since the application can readily take advantage of a new technology or a new release of an existing technology.

Does Rapid Developer support XML?

Yes. Rapid Developer vastly simplifies XML development. XML (eXtensible Markup Language) is a W3C standard that is the evolving standard for content delivery and exchange of information on the Internet. The Rapid Developer platform provides the ability to define XML Document Type Definitions (DTD) and XML documents and allows users to directly map these DTDs to business objects within Rapid Developer. These XML documents are used to send and receive XML messages to publish information, send information, or to receive information from other applications and sites.

What makes Rapid Developer such a productive development environment?

Rapid Developer is the fastest way to create enterprise-class, n-tier applications. Rapid Developer uses architectural tools that define the business model, transaction model, user interfaces, integrations, and the application's deployment configuration in a language-neutral, technology-independent approach. Rapid Developer-created applications are inherently agile, cross-platform and multi-channel, and are architected, not programmed; engineered, not hand-coded; and constantly evolving, never static.

Rapid Developer's dramatic productivity gains are enabled through:

- *A seamlessly integrated development environment, from design through deployment*
- *Model-driven, visual development and RAD techniques, including innovative RAD models*
- *True object-oriented technology*
- *Automated code construction for all tiers of an n-tier system*
- *Automated optimization and deployment for all leading technology platforms (application servers, databases, browsers and messaging transports)*
- *Easy application maintenance for business/functional and technology/platform changes*

The Rational Rapid Developer architected rapid application development environment is comprised of three complementary and completely integrated systems:

- *Rational Rapid Developer Modeling System (RRDMS), that includes selected UML models and RAD extensions beyond current UML concepts, to create a detailed, end-to-end model of your entire application.*
- *Rational Rapid Developer Construction System (RRDCS), that automatically converts your detailed application model into high quality, n-tier deployed and executable code (not just code stubs) within minutes.*
- *Rational Rapid Developer Framework (RRDF), that consists of a set of runtime components supporting basic application, messaging and integration functionality.*

How does Rapid Developer provide protection against technology obsolescence?

Rapid Developer uses model driven architecture (MDA) to create end-to-end application models that are abstracted above specific technology selections. Rapid Developer automatically translates your application model into a high-quality application for your selected deployment technology. Rapid Developer TechBlades™ are code construction templates that automatically optimize Rapid Developer-generated code for specific deployment technologies, including leading application servers, databases, messaging transports, browsers and security. By employing these interchangeable TechBlades, Rapid

Developer lets developers select deployment technologies with the click of a mouse to create applications optimized for each technology. Rapid Developer-based applications are protected from technology obsolescence since the application can readily take advantage of a new technology, or new version of an existing technology, by leveraging the associated TechBlade to automatically regenerate the application with optimized support for the new technology.

Specifically, what code does Rapid Developer generate?

Most n-tier enterprise applications contain the following types of code:

- *Presentation tier - HTML, JavaScript, JSPs, Servlets*
- *Business tier - EJBs, classes, transactions*
- *Data tier - Data Definition Language (DDLs), schemas, stored procedures, sample data*
- *Custom Business Logic – Specific business rules (written in Java)*

From the above listing of code types, Rapid Developer generates 100% of the first three types – with the ability to override and customize presentation and business tier logic. Also, it accelerates development of typically 30% of the custom business logic by utilizing a set of 100-plus pre-built code templates that capture essential application functions. Further, Rational Rapid Developer deploys to a standard, out-of-the-box MVC-2 framework, optimized for a broad range of vendor technologies. Typically, Rapid Developer generates in excess of 95 percent of all code for a typical executable application. Additionally, Rapid Developer insulates developers from the need to know n-tier architecture. This is a key feature given the general shortage of Java programmers and the more acute shortage of enterprise-level J2EE developers. With Rapid Developer, Java skills are focused on creating custom business logic, as well as the general (non-Rapid Developer-specific) intricacies of deploying n-tier applications.

How does Rapid Developer compare to previous CASE tools?

The previous generation of CASE tools could be characterized as:

- *Not object-oriented – For the most part, they generated mainframe COBOL and COBOL II code.*
- *Not RAD – They were overly analytic, requiring exhaustive front-end analysis and design projects.*

- *Not integrated – They required a number of tools (often from competing vendors) in order to function at all.*
- *Not open – They were technology-exclusive, not inclusive in their interfaces or modeling facilities, and tied to a single platform (resulting in premature shelf-life).*

On the other hand,

- *Rapid Developer is object-oriented, RAD, integrated and open.*
- *In addition, Rapid Developer provides:*
- *Development productivity – Productivity through RAD modeling and automated code construction*
- *Scalability – Constructing presentation and persistence classes applies software engineering principles to the generated code and allows for, in many cases, better performing code than could be handwritten*
- *Investment protection – Only code-constructed applications can be technology-neutral*
- *Reuse – Rapid Developer generates pure object-oriented Java code, which can both integrate with and be integrated into any existing application that provides an API-level interface.*

How do I modify the application once it is generated?

Once applications are generated, modifications are made at the model level, not on the code. This dramatically reduces the maintenance effort because the maintenance of models is significantly easier than the maintenance of code. After the changes have been made to the models, Rapid Developer regenerates the application and the old code is overwritten.

Can I override the automatically generated code?

Yes. Rapid Developer incorporates a feature known as code generation override (system method override). This feature allows the developer to preview the automatically generated code. If the developer wants to modify, delete or change any of this code, he/she can make modifications to the code within the tool. The code can then be generated with the desired modifications. These modifications are "remembered" for subsequent regenerations of the application. Note that IBM Rational does not recommend that developers make changes to the code outside of Rapid Developer as modifications to the

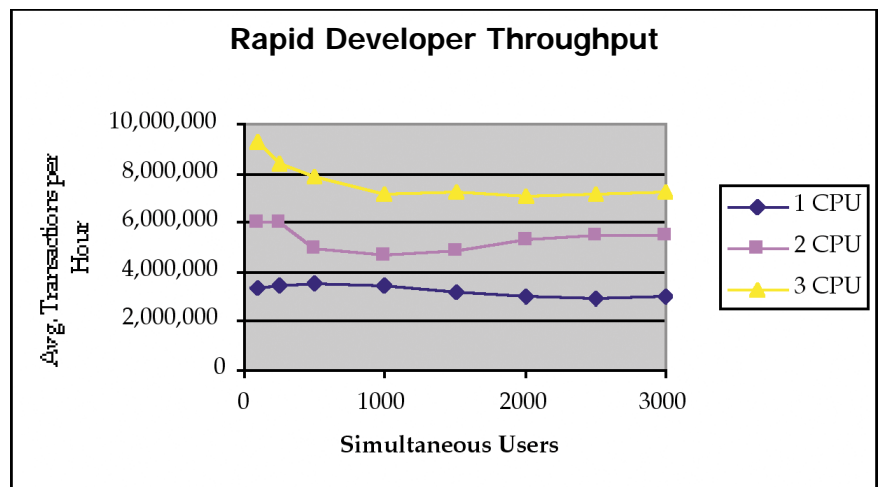
generated code will require future maintenance of the code rather than the models. Maintenance at the model level is one of the key advantages to developing in the Rapid Developer environment because of the added flexibility, speed and inherent portability to new technology platforms it brings to the development and maintenance application life cycle. It is important to note that with Rapid Developer, developers have a 100% open environment and can reach outside of Rapid Developer and invoke any existing logic, components and systems.

Do Rapid Developer-created applications perform?

Rapid Developer-generated code is high quality, high performance code. It conforms to industry standards and is well structured and completely readable.

- *One evaluation by a Fortune 50 client called it “by-the-book.”*
- *Another evaluation of Rapid Developer-generated code by a global systems integrator in a live project, side-by-side test versus hand coding found the Rapid Developer code performed “significantly better.”*
- *A recent third-party performance and scalability test on a Rapid Developer generated “Pet Store”-like application demonstrated scalable and high-performing code. See Figure 8 below.*

Figure 8: Performance & Scalability



Rapid Developer leverages its unique transaction modeling and flexible construction patterns to ensure high quality and performance. In fact, in constructing presentation and persistence classes, Rapid Developer applies

software engineering principles to the generated code that in many cases allows for better performing code than can be handwritten. The transaction model mediates all transactions between actors and business objects; stores its transient data in the transactions ObjectSpace; enforces business logic specified in business objects; coordinates all data flow across tiers in an n-tier system; and provides the basis for high performance and scalable construction engineering. The transaction model only contains those business objects, methods, and attributes required by the transaction. The developer has complete control as to what is projected into the transaction model ensuring it is highly optimized.

Construction patterns provide the developer an ability to optimize their application construction based on specific requirements. This could be based on corporate standards, J2EE blueprints, or application function. For example, transactional Web pages could be constructed to a high performance session bean pattern, and a read-only information page may be constructed to a servlet-only pattern.

For more information, see "IBM Rational Rapid Developer, Scalability and Performance" technical paper.

For more information, see "IBM Rational Rapid Developer, Scalability and Performance" technical paper.

What is ObjectSpace?

Rapid Developer's ObjectSpace is a patent-pending facility that allows rapid transaction modeling. Transaction modeling typically represents more than 80 percent of the effort in a complex business application, and normally involves extensive hand coding. Transaction modeling has become even more complex due to the increased complexity of n-tier architectures and the multitude of delivery channels, including: Web (HTML), wireless (WML), messaging (XML), and Web services (SOAP).

ObjectSpace uses a visual environment for transaction modeling and ties those models directly to the business objects and databases on one side, and to various presentation formats on the other.

Conclusion

We hope this test drive gave you a good overview of what Rational Rapid Developer can provide to application developers.

Rational Rapid Developer enables software developers to build better software faster by extending their development environment with essential, fully integrated tools. This revolutionary tool from the leader in software development combines model-driven development, RAD techniques and automated construction across all tiers to provide the fastest and easiest path to J2EE.

Rational Rapid Developer improves the way you work in the following ways:

Smarter, Faster J2EE Development - Only Rational Rapid Developer provides a single, integrated, architected, rapid application development that empowers a broad class of developers, including 4GL, PowerBuilder, Visual Basic, Oracle Forms, COBOL and others on your staff, to apply their existing application development skills and domain expertise and create robust J2EE applications – with little or no prior J2EE experience.

Experience Architected RAD:

Design. Integrate. Auto-Construct. Deploy. Maintain. Fast. - Provides the richest and deepest set of features and functionality for rapidly designing, constructing, previewing, deploying and maintaining business applications. From start to finish, this single seamless development environment provides the broadest set of features to speed your development projects to completion.

Rational Rapid Developer is the only business application development tool to combine RAD modeling, automated construction and agile application maintenance capabilities into a seamless integrated development environment.

Expedite work through agile, iterative development - Supplies the most productive working environment for developers creating and maintaining business applications. Rational Rapid Developer is the only business application development tool to combine RAD modeling, automated construction and agile application maintenance capabilities into a seamless integrated development environment. This unique combination enables developers to get projects done faster by automating routine development tasks and simplifying application maintenance to a few mouse clicks.

Adapt to platform change with ease - Only Rational Rapid Developer provides developers with a separation across all tiers of application business

functionality from the underlying vendor specific deployment technologies. Once the application is fully modeled and ready to construct, Rapid Developer offers the developer the broadest menu of deployment technology choices to auto-construct fully optimized, technology specific code from the models for deployment. The end result is that migration of your application is as simple as reconstructing to a new technology or a newer version of your existing technology. Additionally, application maintenance can be focused exclusively on business functionality while leaving the headache of constant technology churn to Rational.

Transform your legacy assets - Rational Rapid Developer provides the most robust legacy integration capabilities in a single Java development environment. Whether your requirements include mainframe database connectivity, CICS visual and non-visual transaction integration, Web services, or XML message mapping, Rapid Developer has it covered. Coupled with these robust, enterprise-ready integration capabilities is the user-friendly, visual RAD experience allowing the creation of n-tier applications without needing to know the intricacies of J2EE. The end result is powerful and rapid solution that addresses the multi-channel, composite application environment common in enterprise IT today.

Lastly, Rational Rapid Developer is created and supported by IBM Rational Software, a world-class leader in helping organizations create business value by improving their software development capability. If this test drive has not quenched your thirst for Rational Rapid Developer, here are additional places where you can learn more:

The IBM Rational Web site at <http://www.rational.com/products/rrd/index.jsp>

If you are a current Rational customer with an active maintenance agreement, then you can explore the Rational Rapid Developer Center in the Rational Developer Network (select **Rational Tools >Rational Rapid Developer**).



© IBM, the IBM logo, and WebSphere are trademarks of International Business Machines Corporation in the United States, other countries, or both.

Rational, Rational Unified Process, RequisitePro, and XDE are trademarks or registered trademarks of Rational Software Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product or service names may be trademarks or service marks of others.

Rational Software Corporation is a wholly owned subsidiary of IBM Corp.

© Copyright Rational Software Corporation, 2003. All rights reserved.

Made in the U.S.A.