What's new in Borland® JBuilder® 2005

An overview of the new functionality

in the leading Java™ development solution

A Borland White Paper

September 2004

Borland®

Contents

Introducing Borland® JBuilder® 2005

Borland® JBuilder® 2005 is the leading cross-platform environment for building industrial-strength enterprise Java™ applications. JBuilder 2005 Enterprise simplifies Web and Enterprise JavaBeans™ (EJB™) development with two-way visual designers and rapid deployment to Java™ 2 Platform, Enterprise Edition (J2EE™) application servers. JBuilder powers developer productivity with innovative JavaServer™ Faces (JSF), Struts, and Web Services designers, distributed refactoring, and integrated performance tools.

This white paper takes a look at some of the new and enhanced development tools in JBuilder 2005 and how they can boost your team's productivity and accelerate the application lifecycle.

The developer's gateway to ALM

JBuilder is the developer's gateway to the application lifecycle for the design, development, testing, deployment, and management of Java applications. JBuilder provides a platform in which to manage the development lifecycle by tightly integrating with Borland® Optimizeit™ Suite for performance management, Borland® CaliberRM™ requirements management, Borland® StarTeam® for change management, Borland® Enterprise Server for J2EE, Web, and CORBA® deployments, and—shortly after the release of JBuilder 2005—Borland® Together® technologies for modeling and analysis. JBuilder also integrates with a wide variety of leading application servers and other tools.

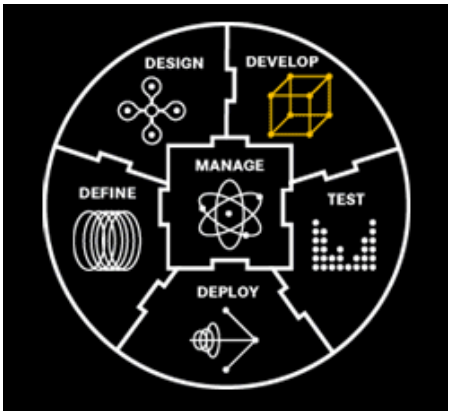

Figure 1: *JBuilder is the "develop" piece of the Borland ALM solution for Java*

What's new in JBuilder 2005

The new capabilities in JBuilder focus on several key areas important to Java development teams. JBuilder 2005 helps you:

- Build Web interfaces faster
- Speed development by taking advantage of the latest Java technologies
- Build higher-quality, higher-performance applications
- Increase team efficiency
- Benefit from expanded integration

Build Web interfaces faster

Take advantage of new JavaServer Faces (JSF) technologies to build and connect Web interfaces to your applications faster without losing control over your code. The new drag-and-drop visual flow designer and JSF code editor make it easy to create JSF artifacts and assemble them into a fully functional application. You'll be amazed by how quickly you can jump-start building Web interfaces with JSF.

JSF Flow designer

The JSF Flow designer is a JavaServer™ Pages (JSP)-centric visual designer for working with JSF navigation rules and navigation cases. For JSPs in a JSF application, note the new Flow page, which displays—and allows you to manipulate—the relevant navigation rules within the context of that particular JSP.

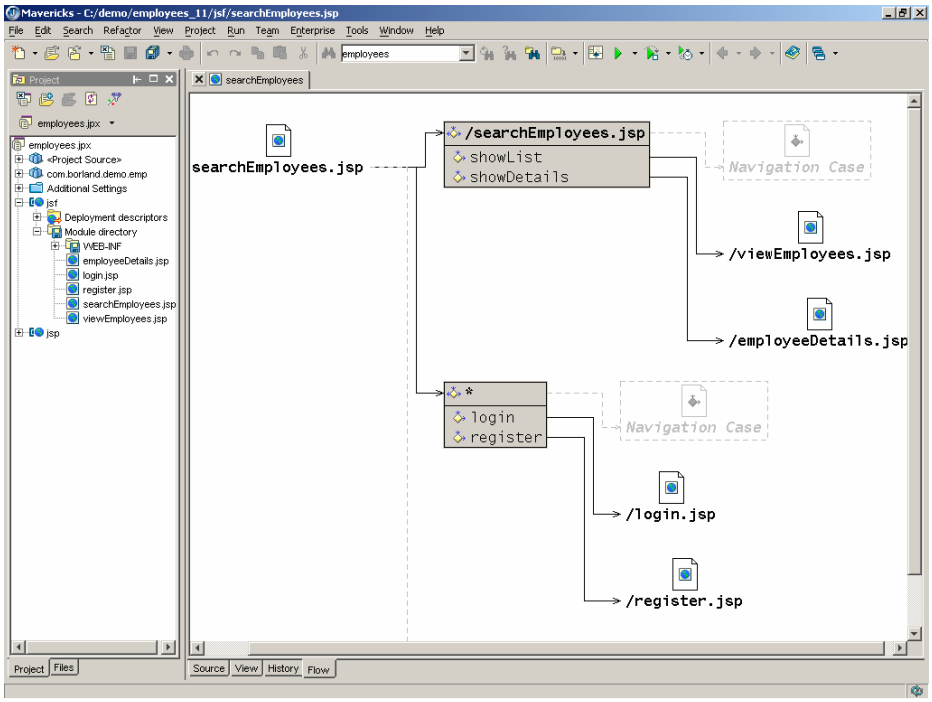

Figure 2*: JSF Flow designer in JBuilder 2005*

JSF Configuration Editor

The JSF Configuration editor is a visual editor for the JSF (faces-config.xml) deployment descriptor. To open the editor, double-click the faces-config.xml file in the WEB-INF node of your module directory in the project pane.

JSF coding wizards

For certain properties of JSF components, such as the action property of a Command Button, wizards for setting the property are available from context menus in the tag inspector and from ErrorInsight.™ For example, the wizard for the action property of a Command Button allows you to create the action's method binding.

Speed development with the latest Java technologies

Step up to the benefits of Java™ 2 Platform, Standard Edition (J2SE™) 5.0 and J2EE 1.4 to speed development and enhance scalability and performance. JBuilder 2005 provides support for all the new language features of J2SE 5.0 (formerly known as JDK™ 1.5) and support for J2EE 1.4 with XML, Java, and Web Services. A capability of JBuilder 2005 not found in any other product is that the compiler can compile source code using some of the 5.0 language features to be 1.4 binary compatible, so that developers can start using 5.0 language features even if their application will run on a 1.4 JVM.

JDK 1.5 refactorings

JBuilder now supports JDK 1.5 refactorings that update code to use JDK 1.5 language features. The following refactorings for JDK 1.5 code are supported:

- Foreach refactorings: Foreach refactorings refactor existing loops to JDK 1.5-style enhanced loops. You can refactor the following types of loops:
    - Array traversal
    - List traversal
    - Iterator *for* loops
    - Iterator *while* loops
- Auto(un)boxing refactorings: The autoboxing and auto-unboxing features in the JDK 1.5 allow you to easily convert between primitive types and their object-based counterparts.
- Generics refactorings: JDK 1.5 introduces the use of Generics. Generics add compile-time type safety to the Collections API and eliminate casting. In JDK 1.4 and earlier, an object had to be cast to the appropriate type before use. With Generics, the need for casting is eliminated.

New Web Services wizards

There are four Web Services configuration wizards on the Web Services page of the object gallery. Each toolkit has two wizards — the Web Services Server Configuration wizard and the Web Services Client Configuration wizard.

Server configuration: Use the Web Services Server Configuration wizard when you want full Web Services designer functionality for exporting and importing services.

Client configuration: Use the Web Services Client Configuration wizard when you only want to import services.

Module Chooser wizard: The Module Chooser wizard allows you to create or choose a Web Services module (client or server) when using these project pane context menus to export and import Web Services — Export As A Web Service and Import As A Web Service. The Module Chooser wizard might also be launched when importing a WSDL from UDDI using the Web Services Explorer.

Web Services Interoperability (WS-I) testing tools

The Web Services Interoperability Organization (WS-I) provides assistance to Web Services developers for meeting interoperability standards for Web Services. WS-I clarifies existing specifications and provides testing tools and sample applications. JBuilder supports the following WS-I tools:

- The **WS-I Monitor tool** sends messages on to some other end point while preserving the integrity of communication between the two end points and records the messages that flow through it to a log file.
- The **WS-I Analyzer tool** validates messages sent to and from a Web Service and verifies the description of the Web Service. This includes the WSDL document that describes the Web Service and the XML schema files that describe the data types used in the WSDL service definition.

Enhanced mobile development

Mobile development with JBuilder includes several new features and changes. JBuilder mobile development includes Java™ 2 Platform, Micro Edition (J2ME™) Wireless Toolkit Version 2.1 with support for MIDP 2.0 and 1.0 and for CLDC 1.1 and 1.0. JBuilder has added

support for designing a DoJa 3.0 UI in the UI designer. The MIDlet Archive Builder now includes a step for digitally signing your MIDlet Suite. Unit testing has been expanded to include mobile development. A new micro test runner has been added to the runner type list in the New Runtime Configuration dialog box. Two new wizards have been added.
The Micro Test Case wizard creates a skeleton test case class for unit testing a class in a mobile project. The Micro Test Builder wizard creates a test suite that groups test cases so that they can be run as a batch.

Build higher-quality, high-performance applications

Borland® Optimizeit™ Profiler is now included at no extra cost in JBuilder 2005 Developer. JBuilder 2005 Enterprise includes the entire Borland® Optimizeit™ Enterprise Suite, which enables you to analyze the behavior of code across J2EE application tiers with a Java profiler, thread debugger, code coverage, and Borland® Optimizeit™ Request Analyzer for J2EE profiling.

JBuilder developers can now track performance bottlenecks at the JDBC,™ JMS, JNDI,™ JSP, and EJB level, drilling down to the exact line of source code for root-cause precision. Optimizeit Enterprise Suite equips J2EE developers with a comprehensive toolkit to optimize performance and manage application quality throughout development.

Code audits

Code audits help you discover potential problems early in the development cycle, preventing more-complex code problems later. Just turn on audits inside the project properties, and when a compile is done, the warnings show in the Structure pane like errors and todo's. Clicking on the warning takes the developer to the particular code statement or block.

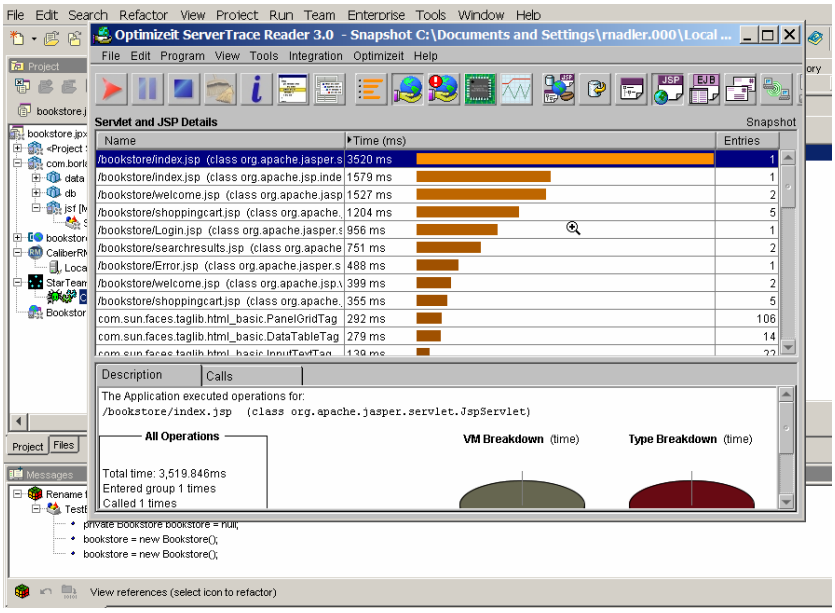

Figure 3: *Using JBuilder with Optimizeit to identify potential performance issues in a J2EE application*

Optimizeit Profiler

JBuilder Developer and JBuilder Enterprise now include Optimizeit Profiler. Isolate memory and CPU performance issues using the advanced capabilities of Optimizeit Profiler to save time and effort. Feature highlights include the Automatic Memory Leak Detector, Automatic Application Quality Analyzer, and Progress Tracker.

Optimizeit Thread Debugger

Easily solve challenging thread issues with Optimizeit Thread Debugger. View the status of all threads and monitors in real time, avoid thread starvation and contentions that lead to crashes, and predict deadlocks before they occur.

Optimizeit Code Coverage

Borland® Optimizeit™ Code Coverage gives developers confidence that their code is ready to deploy when they are running performance checks during development.

Optimizeit Request Analyzer

Optimizeit Request Analyzer provides advanced profiling techniques that allow developers to analyze the performance behavior of code across J2EE application tiers. Using Optimizeit Request Analyzer, developers can efficiently prioritize the performance of JDBC, JMS, JNDI, JSP RMI, and EJB Web requests so that trouble spots can be proactively isolated earlier in the development lifecycle. Precise drill-down capabilities accelerate the time-to-resolution of performance problems, diagnosing and locating the root cause right down to the offending line of source code.

Increase team efficiency

JBuilder 2005 helps increase your team efficiency with distributed refactoring that enables you to refactor across libraries or projects, Apache Ant script debugging support, improved version control integrations, and support for Subversion.

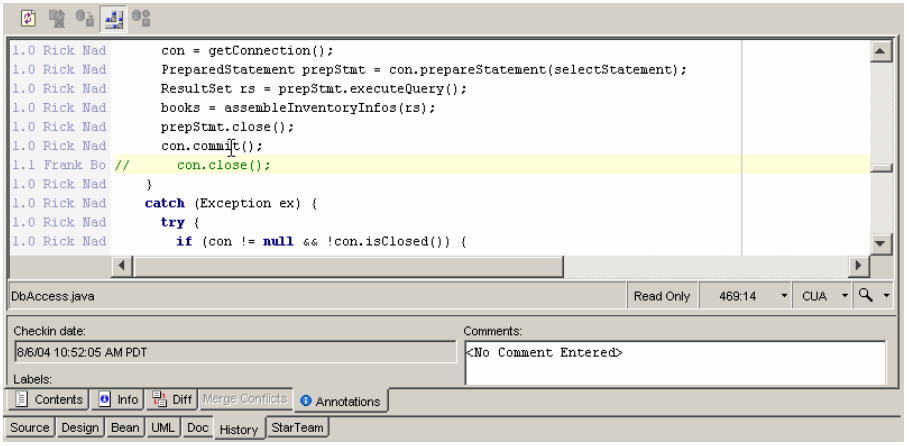

Figure 4: *Use the history browser to see which team member last changed each line of code and any comments entered*

Distributed refactoring

JBuilder now records completed refactorings, enabling you to track the history of refactorings and to share refactorings among team members. You can add this history to the project archive, making the history available to other projects and libraries that were not available in

the initial refactoring. Use the history to easily update projects that depend on refactored external APIs, SDKs and libraries.

To use distributed refactoring, you must first add the refactoring history to the depended-upon project or library's archive. When you open that depended-upon project or library archive from the dependent project, you can then review all refactorings with the Refactoring History. You will first refactor your project globally, so that all files in the project are updated to the new code symbols. You can then use ErrorInsight to refactor any remaining instances of out-of-date code.

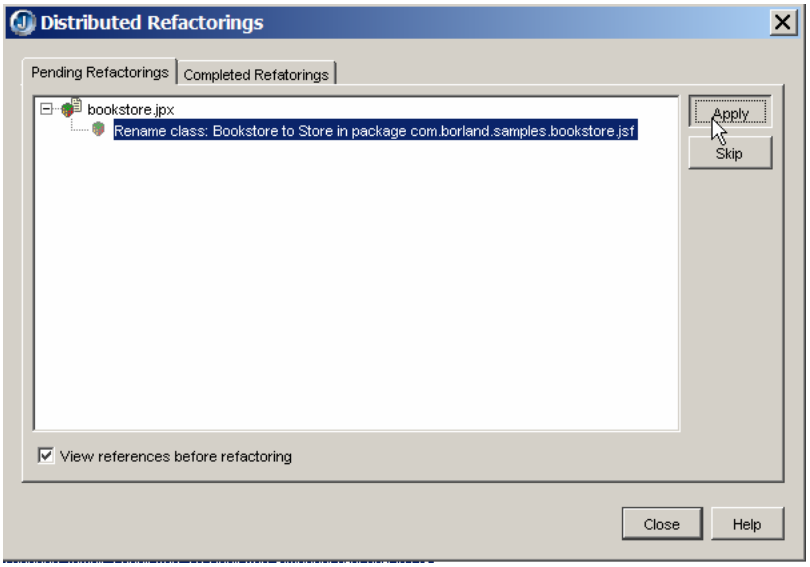

Figure 5: *Applying a refactoring to a dependent project using distributed refactoring*

Ant debugging

You can now debug Ant build files. Set breakpoints inside an Ant build file, inspect Ant properties, and step between tasks. To debug an Ant build file, open your Ant file in the editor, set a breakpoint, right-click the Ant build file in the project pane, and choose Debug. The debugger stops execution at the breakpoint. When execution is paused, you can step into a target or task.

Subversion integration

JBuilder 2005 supports Subversion, an open-source tool originally derived from the CVS project. Subversion is particularly useful to server administrators and is gaining popularity. JBuilder supports most common Subversion tasks from within the IDE.

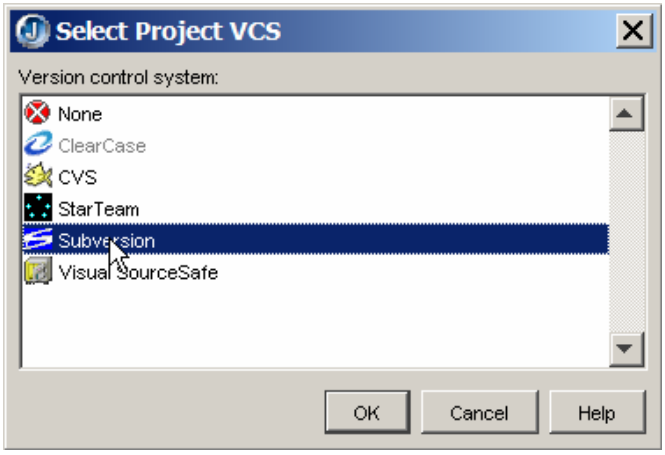

Figure 6: *Subversion is a new choice for team development in JBuilder 2005*

Improve application security

As J2EE and Java Web development have been adopted increasingly by the mainstream, awareness of its vulnerabilities have become more widespread. Because a typical distributed Java application is usually built by many developers focusing on different abstraction layers of a system, it becomes more difficult to understand where a system is exposed. For instance, who is responsible for verifying that an input field is being used for a database lookup is actually a value rather then an SQL statement? How many Java developers today have been trained in security?

The exposure of vulnerabilities increases as applications are exposed outside of the corporate firewall. As antipatterns in the realm of code stability become better known, security antipatterns are still foreign to many developers, because security is a quality aspect that often is still addressed in a reactive manner (monitoring) rather then proactively (security audits).

Borland JBuilder 2005 includes Fortify Software for JBuilder, which provides a set of security audits that help developers discover antipatterns specifically related to security vulnerabilities.

Benefit from expanded integration

Benefit from expanded integration across the application lifecycle. Direct integration with Borland® CaliberRM™ requirements management lets you view, link, and filter requirements right from within the JBuilder IDE. Enhanced integration with Borland® StarTeam® provides support for project groups. Plus, bundled premium tools expand your development toolkit even more.

CaliberRM integration

JBuilder 2005 includes a new plugin for CaliberRM, a popular requirements management system from Borland. The plug-in allows you to view requirements directly within the JBuilder IDE:

- Create a connection to a CaliberRM server and log in or log out
- View a CaliberRM project and baseline
- Filter requirements to see only the ones you want
- Link requirements to your source code for bi-directional navigation
- Update comments for requirements that have been changed on the server

You can also change to a new server connection, project, and baseline as you work. Additionally, you can open the full CaliberRM client from the plugin. Once a requirement has been traced to a source file, you can open that source file in the editor.

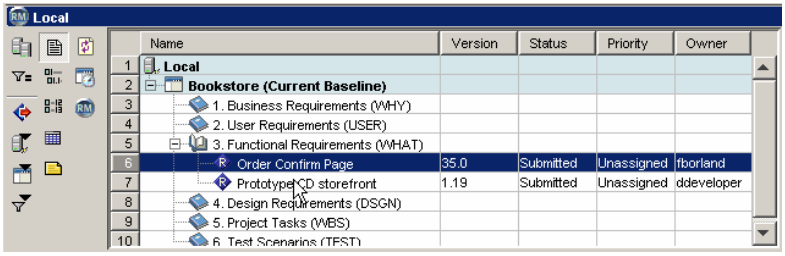

Figure 7: *Browsing CaliberRM requirements in the JBuilder IDE*

Enhanced integration with Borland® StarTeam®

This release of JBuilder supports StarTeam versions 5.4 and 6. The folder tree pane, hosted in the structure pane, provides tabs that allow you to see the Project, Shortcut, View, or Server tree. Shortcut support allows you to create shortcuts for folders, files, change requests, and other items. You can store these shortcuts wherever you like. Doing so lets you keep all of your shortcuts in the project pane view most convenient to you and allows you to work on your material in any other project. Use the icons in the toolbar at the top of the upper pane in the StarTeam view to create these shortcuts. Use the context menus on the shortcuts themselves to rename, move, and edit them.

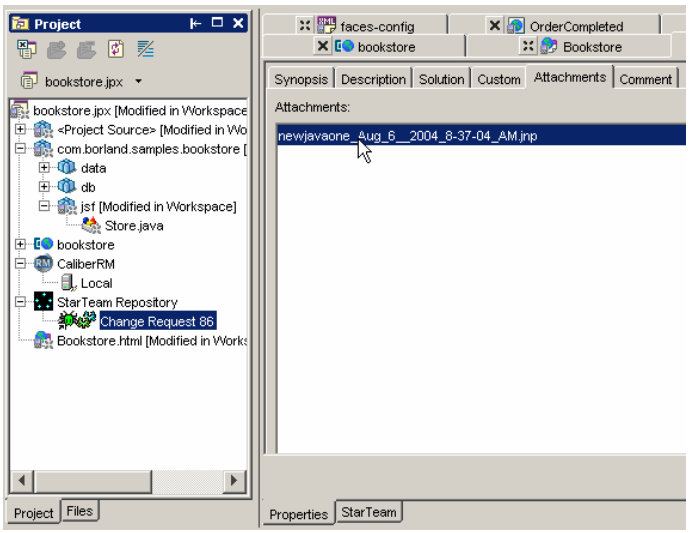

Figure 8: *Viewing a change request from StarTeam*

Expanded selection of Premium Tools

Crystal Reports® for Borland® JBuilder®

Tightly integrated with JBuilder, the custom version of Crystal Reports provides data connectivity, presentation, and integration components to simplify and accelerate the process of developing your application's JSP presentation layer.

eBay™ SDK for Java™

Today, more than $23 billion in transactions take place globally on the eBay™ Platform. The eBay™ Software Development Kit (SDK) for Java™ —a package of libraries, sample applications, and documentation that can significantly cut time-to-market of new eBay applications—makes it easier to integrate an application, tool , or service with the eBay Platform.

Fortify® Software for Borland® JBuilder®

Find security vulnerabilities in your Java code so you can fix them quickly and easily. Fortify Software's unique source code security analyzer and secure coding rules for Java and J2EE locate security flaws in your code accurately and automatically.

Sybase® EAServer integration module for JBuilder

Provides deployment and debugging of J2EE applications from JBuilder into Sybase EAServer.

Vignette® Portal Plugin for Borland® JBuilder®

Provides creation wizards, sample implementations, server integration, and complete documentation to help developers quickly create, deploy, and test JSR 168 portlets.

JBuilder editions

JBuilder 2005 is available in multiple editions to suit the needs of different Java developers and budgets.

JBuilder Enterprise

JBuilder Enterprise is a comprehensive development solution for designing, coding, and deploying industrial-strength enterprise Java applications. Development teams who want to build EJB applications or Web Services, want easy deployment to J2EE application servers, and want tools and wizards to speed development and optimize performance choose JBuilder Enterprise for its powerful enterprise development and deployment capabilities.

JBuilder Developer

JBuilder Developer is a powerful Java IDE that speeds development of Web, database, and mobile applications. Developers and teams looking for a lower-cost, lighter-weight yet robust development solution choose JBuilder Developer for its powerful features, including visual designers for JavaServer Faces and Struts, distributed refactoring, code audits, unit testing, advanced debugging, team development, and included Borland Optimizeit Profiler to deliver fast, scalable, reliable applications.

JBuilder Foundation

JBuilder Foundation is a great Java development environment for students who want to learn Java and for developers who want a redistributable foundation on which to build standalone Java applications and plugins.

The feature matrix below compares the new features in JBuilder Developer and Enterprise. For more details about all of the JBuilder 2005 features, see the full feature matrix on the JBuilder product page at http://www.borland.com/jbuilder.

JBUILDER 2005 NEW FEATURE COMPARISON	DEVELOPER	ENTERPRISE
JavaServer™ Faces editor and visual flow designer	X	X
J2SE™ 5.0 support, refactoring, and debugging	X	X
Optimizeit™ Profiler	X	X
Code audits	X	X
Distributed refactoring	X	X
Enhanced version control support and Subversion integration	X	X
Enhanced Web development tools	X	X
Ant script debugging	X	X
Optimizeit™ Request Analyzer for J2EE™ profiling		X
J2EE 1.4-enabled EJB™ designer		X
Incremental deployment to J2EE application servers		X
Enhanced Web Services development		X
Direct link to requirements in CaliberRM™		X
Premium tools: Crystal Reports® for Borland® JBuilder,® Fortify® Software for Borland® JBuilder,® Sybase® EAServer plug-in, eBay™ SDK for Java,™ and Vignette® Portal plug-in for Borland® JBuilder®		X

Summary

As you can see, JBuilder 2005 is a compelling upgrade for current JBuilder users and a great choice for development teams looking to standardize on a development solution. We've examined how JBuilder 2005 can help developers build Web interfaces faster using JavaServer Faces. JBuilder promises to speed development by taking advantage of the latest Java technologies including J2SE 5.0 and J2EE 1.4. JBuilder enables development teams to build higher-quality, higher-performance applications with the included Borland Optmizeit Profiler and performance tools. Teams can become more efficient through sharing refactorings and additional version control support. And, finally, development teams using JBuilder 2005 will benefit from expanded integration across the Borland ALM solution with CaliberRM and StarTeam as well as other application servers, version control systems, and Java technologies—all in all, enabling software development teams to deliver better software faster.