InfoWorld
GET TECHNOLOGY RIGHT

IT Strategy Guide

Java IDE Comparison Strategy Guide: InfoWorld's Java IDE of the Year Takes on the Best That the Industry Has to Offer

INSIDE

Compliments of:

EMBARCADERO TECHNOLOGIES®

Introduction

When it comes to development environments for creating Java-based applications, there are two words to keep in mind – Eclipse and JBuilder. The first leverages the work by open source communities to build an extensible tools platform for software development. The second leverages Eclipse to create commercial-grade tooling for the enterprise. Eclipse is both a phenomenon and an incredible success story. The Eclipse Foundation and the Eclipse Project have become standards-bearers both for open source in general, and software development specifically.

The Eclipse Foundation is incredibly vibrant and successful, and it has created the de facto standard platform for building Java software. Dozens of top-rated software and hardware companies, from IBM to Intel, from Oracle to SAP, from CA to Borland, drive the Eclipse Foundation forward.

... for every dollar spent on JBuilder, an organization can expect a return of $90-$165 in savings through increased developer productivity and improved software quality.

The Eclipse tools platform can be used on its own as a free open-source development environment, but it also serves as the foundation for many commercially available Integrated Development Environments (IDE). Companies using the Eclipse framework as the foundation of their product offerings include IBM, Genuitec and CodeGear. Of course, there are also tools and platforms available to enterprise Java developers which are not based on Eclipse, but as this IT Strategy Guild shows, the momentum is clearly behind the Eclipse-based solutions.

If you're looking for a commercial software development platform for Java, you want one based on Eclipse. But which commercial offering do you want? We'll go deep into the details, comparing CodeGear JBuilder 2007, IBM Rational Application Developer (IRAD) and Genuitec MyEclipse against the free open-source Eclipse platform. The results are startling and impressive: the leader, JBuilder 2007, boosts productivity significantly. In fact, for every dollar spent on JBuilder, an organization can expect a return of $90-$165 in savings through increased developer productivity and improved software quality.

What about Sun? It's easy to think that Sun – which created Java, and manages the Java Community Process — would have a natural advantage when it comes to development tools. However, the reality of the situation, as verified by the InfoWorld Test Center, proves otherwise. Our in-depth review shows how Sun NetBeans 5.5 fared against the two biggest names in Java: IRAD 7.0 and JBuilder 2007 Enterprise Edition. As the reviewer said, "JBuilder ... is a truly standout IDE."

We close this IT Strategy Guide with a look at an advanced concept in Java development: Application Factories. We'll show how these models transfer knowledge between team members, so that everyone can understand the intent of the architects, interface designers and other senior developers throughout the organization. Application Factories are a new approach to software development and code reuse. This innovative development metaphor and associated collection of tools allows developers to focus more on the nature and purpose of the application, and less on the underlying platform, framework, and technologies being used.

— *Alan Zeichick*

Java Development Productivity and Quality Using Eclipse:

A Comparative Study of Commercial Eclipse-based IDEs

The productivity benefits of using commercial Eclipse-based Java IDE products from IBM (IBM Rational Application Developer), Genuitec (MyEclipse), and CodeGear (JBuilder) compared to the freely downloadable baseline Eclipse configuration.

Report Prepared by CostXperts

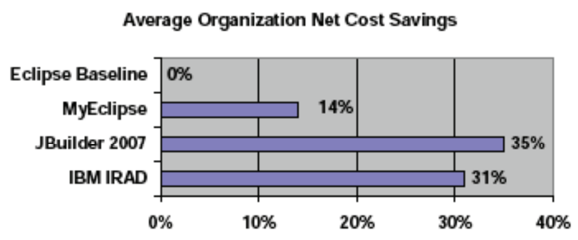
The Cost Xpert Group, Inc. (www.CostXpert.com) specializes in software metrics and predictive models. Considered one of the top experts in the field of software development cost analysis our services are focused on helping clients substantially increase the probability of completing software development projects successfully.

Our staff combines real-world software project management experience with technical training in areas such as parametric cost analysis, system dynamic modeling of software processes, knowledge based modeling of risk, and both stochastic and deterministic optimization of project operations. Many of our consultants are world-renowned leaders in their field of expertise. The CostXpert Group has over 5,000 customers including Boeing Corporation, Chevron Information Technology, Ernst & Young, Hewlett-Packard, and Unisys Corporation

Executive Summary

Eclipse is both a phenomenon and success story in the Java eco-system and in the overall software development field. More than just a software development tool, Eclipse represents an open source community dedicated to building a development platform and to offering a wide range of extensible frameworks, tools and runtimes for building, deploying and managing software across the application lifecycle.

The attraction of Eclipse can be attributed to many factors including its open source model, flexibility, expandability, extensive commercial industry support, and of course low cost; a baseline Eclipse configuration for Java can be downloaded for free. Many vendors enhance this base configuration with value-added technologies and services for which they charge a license or support fee.



For instance, companies such as IBM, Genuitec, and CodeGear (Borland's Developer Tools spin-off) have developed new Java IDE solutions based on Eclipse. Each of these three Eclipse-based solutions has taken a different strategy and approach to enhancing the baseline Eclipse configuration, delivering unique value to Java developers. By comparison, some companies, such as Sun Microsystems and JetBrains, license development tools based on proprietary technologies developed independently from Eclipse's open framework.

The goal of this study was to objectively measure the benefits of using commercial Eclipse-based Java IDE products from IBM® (IBM Rational Application Developer®), Genuitec® (MyEclipse®), and CodeGear (JBuilder® 2007 Enterprise Edition). These benefits are compared to the freely downloadable baseline Eclipse configuration.

In this study, team configurations and projects of varying sizes and purposes were modeled and measured under two scenarios: (1) building new Java software and (2) enhancing/maintaining existing Java applications. The study measured development cost, time to completion, and resulting application quality. In all situations, all three commercial IDEs (MyEclipse, JBuilder, and IRAD) were found to offer substantial development cost savings and project quality improvements over the baseline free Eclipse distribution.

For typical software development organizations, these percentages translate into substantial net hard dollar savings in terms of software development personnel, time and quality. For the representative organizations used in this study, the return on investment (ROI) of acquiring JBuilder ranged from 90:1 to 165:1. That is, for every dollar spent on JBuilder, an organization can expect a return of $90-165 in savings through developer productivity and improved quality.

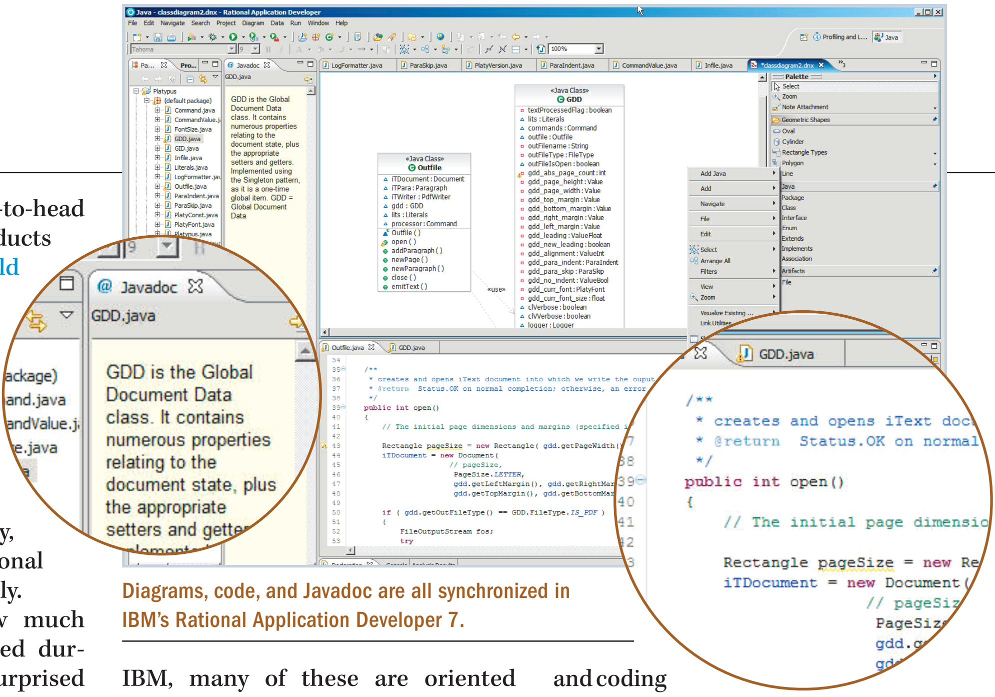
For a free download of this complete report, visit www.codegear.com/products/jbuilder. No registration is required.

IDEs Perk Up

We test a triple shot of Java dev tools: IBM and Borland/CodeGear's Eclipse-based platforms and Sun's open-source NetBeans

BY ANDREW BINSTOCK

JAVA IDEs ARE ONE OF THE MOST USED APP DEV tools in corporate development. They are also among the most capable developer products on the market. With that in mind, it's time to ask yourself: Are you using the Java IDE best suited to your needs, or is it time to re-evaluate?


GDD is the Global Document Data class. It contains numerous properties relating to the document state, plus the appropriate setters and getters. Implemented using the Singleton pattern, as it is a one-time global item. GDD = Global Document Data

```
/**
 * creates and opens iText document into which we write the ouput
 * @return  Status.OK on normal completion; otherwise, an error
 */
public int open()
{
    // The initial page dimensions and margins (specified i

    Rectangle pageSize = new Rectangle( gdd.getPageWidth()
    iTDocument = new Document(
                    // pageSize,
                     PageSize.LETTER,
                     gdd.getLeftMargin(), gdd.getRightMar
                     gdd.getTopMargin(), gdd.getBottomMar

    if ( gdd.getOutFileType() == GDD.FileType.IS_PDF )
    {
        FileOutputStream fos;
        try
```

Diagrams, code, and Javadoc are all synchronized in IBM's Rational Application Developer 7.

InfoWorld last did a head-to-head comparison of these products in March 2005 (infoworld.com/2677) and since then, the IDEs have all undergone important changes. This time, I decided to examine the winner of that review (Borland JBuilder) plus the winners of *InfoWorld*'s Technology of the Year awards, in the Java IDE product category, for 2006 and 2007 — IBM Rational and Sun NetBeans, respectively.

I was impressed by how much these products have matured during the past two years, but surprised that they haven't advanced further in some respects. Other products in the Java developer's toolkit — static code analysis, unit testing, and build management, for example — have seen more progress in this time frame.

IBM Rational Application Developer for WebSphere Software 7.0

Big Blue's entry, colloquially referred to as RAD 7, is based on Eclipse, the open-source software framework that also powers Borland/CodeGear's JBuilder product line. Eclipse was initially a Java IDE, but in recent years, it's been repositioned as a framework into which manifold plug-ins can be added, thereby constructing all sorts of tools (many of which have nothing to do with Java development).

Eclipse's new framework orientation has raised some concerns that the original Java IDE concept has been deprioritized. This perception probably has some validity, but it's offset by the fact that Eclipse currently has the largest community of Java plug-ins currently available.

RAD 7 adds several components to its modified Eclipse base. As befits IBM, many of these are oriented toward enterprise applications. They include portlet and portal development tools, and extensive database support. The database support, which predictably favors IBM's DB2, includes the ability to write user-defined functions (UDFs), stored procedures, and SQLJ code—all unique capabilities among the products reviewed here.

Diagramming and modeling are limited, however: RAD 7 supports only two UML diagrams (versus eight for NetBeans and nine for JBuilder). For fuller diagram support from IBM, you must buy the more expensive IBM Rational Software Architect.

With RAD 7, you can develop Web services and Web service clients, generate WSDL, and even do unit testing against a private UDDI service. In addition, there is support for IBM's DADX, a DB2 XML extension for use by Web services. Alas, the IBM-centricity factors into the UDDI testing as well: It supports only private registries that use IBM-based technologies.

At the coding level, RAD 7 provides a static analysis tool that incorporates more than 200 rules developed by IBM regarding possible Java defects and coding errors. While the other IDEs in this review offer more rules, IBM's solution flagged errors that those productss did not catch. The rules were enhanced by good descriptions of the reasons for the rules and sample code for fixing the problems.

This extensive help reflects IBM's long-standing tradition of great documentation. RAD has links to comprehensive tutorials and IBM's Web site—well known in the developer community for its rich collection of articles—provides additional resources.

If applications require a scripting language to "glue" portions together, RAD 7 has built-in support for Jython (Java-based Python). Unfortunately, the IDE cannot tell automatically what is Jython and what is Java although syntactically the languages are entirely different. This can lead to actions that make the IDE balk.

This flaw, small as it is, reflects a frequent experience I had with RAD 7: many features are not implemented well. For example, installing the software was very difficult. After consid-

I was impressed by how much these IDEs matured during the past two years, but surprised that they haven't advanced further in some respects.

erable support from IBM, I got the product installed correctly, although the original problems were never identified.

There's more. An option to spell-check comments and literals (a useful capability) does not work because IBM ships no dictionary; if the feature is enabled, it marks all words as misspelled. The code-checking tools occasionally prescribe invalid corrections. Dynamic help in dialogs frequently takes you to the wrong level of help, so you're forced to navigate back to your specific context.

Over time, the accumulation of these problems makes this otherwise good product frustrating to use.

I have one other complaint: IBM is far behind the other vendors in supporting existing Java standards. It is the only IDE in this review that has no support for either Java EE 5 or Java SE 6.

I'd recommend RAD 7 to sites already heavily committed to IBM, due to the product's special support for those products, especially DB2 and WebSphere. Sites that want the same IDE for developers in many countries will also like RAD 7, as it is implemented in far more foreign languages than any other IDE. However, the comparatively high price and my other complaints should encourage sites to examine all options before committing their dollars.

Borland/CodeGear JBuilder 2007 Enterprise Edition

JBuilder 2007 garnered first place in our last round-up. This edition is the first release since the product was ported to the Eclipse platform. It is shipped by CodeGear, a division of Borland that focuses on IDE tools.

Due to Borland's well-publicized difficulties and the fact that this is the first release on a new platform, I expected a good product with rough edges. Instead, I found a very smooth, very robust IDE with many innovative features. It's safe to say that CodeGear decided to throw everything it had at this release, and succeeded brilliantly.

For Java coding, JBuilder has three different sets of code auditors and analyzers: the open-source PMD, Findbugs, and Borland's own code-inspection tool. These work well together (in fact, they run the risk of overflowing the developer with flagged items), although they lack actionable explanations of the problem as well as the thoughtful resolution recommendations found in IBM's RAD7 product.

JBuilder bundles a metrics package that is more extensive than any I've seen in any IDE. It generates more than 80 different metrics, displaying them diagrammatically or in spreadsheet format. (Curiously, the metrics do not include the maintainability index, although all the metrics that make up this index are computed.) You can turn off the metrics you're not interested in and set thresholds for those you do want to track. JBuilder also saves metrics snapshots, so that you can compare the current state of the codebase with previous runs to make sure the numbers are trending in the right direction.

CodeGear integrates JBuilder's OptimizeIt suite of tools, which Borland

IBM Rational Application Developer for WebSphere Software 7.0

IBM, ibm.com

GOOD	7.9
Features (40%)	8
Ease-of-use (20%)	8
Integration (20%)	8
Performance (10%)	8
Value (10%)	7

COST: $4,120 (includes 12 months of support)

PLATFORMS: Windows, Linux

BOTTOM LINE: IBM's RAD 7 is a robust, capable IDE that integrates especially well with other IBM technologies and has good visual editors. However, it does not support Java EE 5 or Java SE 6, and it has limited modeling capabilities — two big drawbacks.

Borland/CodeGear JBuilder 2007 Enterprise Edition

Borland/CodeGear, codegear.com/jbuilder

VERY GOOD	8.6
Features (40%)	9
Ease-of-use (20%)	8
Integration (20%)	9
Performance (10%)	8
Value (10%)	8

COST: $1,999

PLATFORMS: Windows (Linux, Mac OS ship in May)

BOTTOM LINE: JBuilder is a smooth, well-designed, capable IDE. It offers excellent metrics and code inspections plus stellar team integration tools. It works seamlessly with numerous open-source tools, Java servers, and databases. It is limited for the moment to Windows only.

Sun NetBeans 5.5

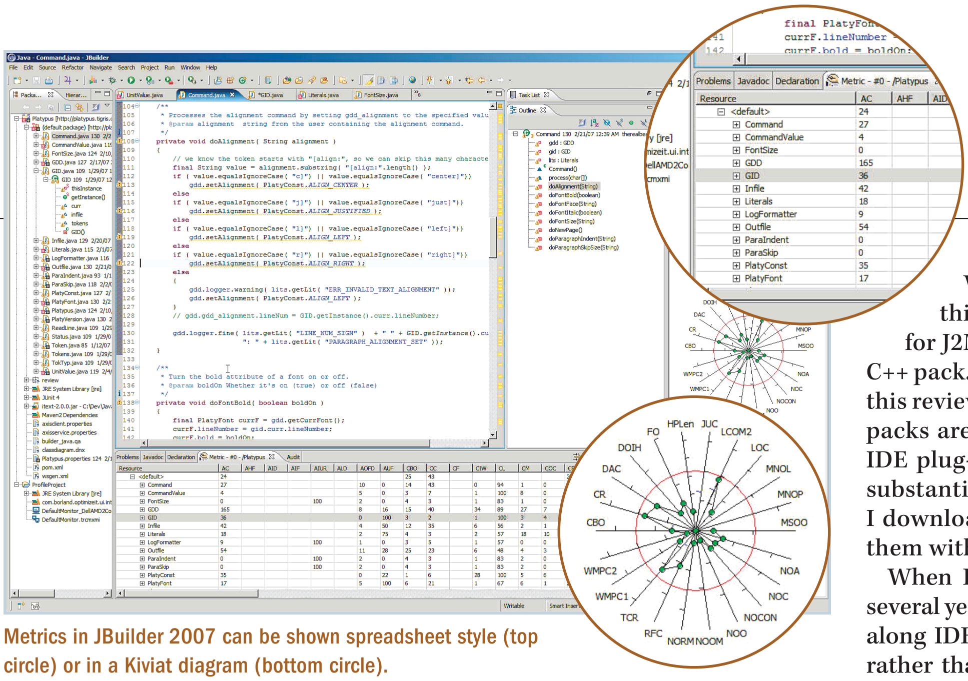
Sun Microsystems, netbeans.org

GOOD	7.4
Features (40%)	7
Ease-of-use (20%)	8
Integration (20%)	6
Performance (10%)	8
Value (10%)	10

COST: Free

PLATFORMS: Windows, Linux, Mac OS, Solaris

BOTTOM LINE: Great collaboration tools and a superior GUI designer distinguish this open-source Java IDE, but missing features (some of which will appear in the imminent 6.0 release) and lack of integration with enterprise technologies diminish NetBeans 5.5.



```
    /**
     * Processes the alignment command by setting gdd_alignment to the specified valu
     * @param alignment  string from the user containing the alignment command.
     */
    private void doAlignment( String alignment )
    {
        // we know the token starts with "[align:", so we can skip this many characte
        final String value = alignment.substring( "[align:".length() );
        if ( value.equalsIgnoreCase( "c]") || value.equalsIgnoreCase( "center]"))
            gdd.setAlignment( PlatyConst.ALIGN_CENTER );
        else
        if ( value.equalsIgnoreCase( "j]") || value.equalsIgnoreCase( "just]"))
            gdd.setAlignment( PlatyConst.ALIGN_JUSTIFIED );
        else
        if ( value.equalsIgnoreCase( "l]") || value.equalsIgnoreCase( "left]"))
            gdd.setAlignment( PlatyConst.ALIGN_LEFT );
        else
        if ( value.equalsIgnoreCase( "r]") || value.equalsIgnoreCase( "right]"))
            gdd.setAlignment( PlatyConst.ALIGN_RIGHT );
        else
        {
            gdd.logger.warning( lits.getLit( "ERR_INVALID_TEXT_ALIGNMENT" ));
            gdd.setAlignment( PlatyConst.ALIGN_LEFT );
        }
        // gdd.gdd_alignment.lineNum = GID.getInstance().curr.lineNumber;

        gdd.logger.fine( lits.getLit( "LINE_NUM_SIGN" )  + " " + GID.getInstance().cu
                             ": " + lits.getLit( "PARAGRAPH_ALIGNMENT_SET" ));
    }

    /**
     * Turn the bold attribute of a font on or off.
     * @param boldOn Whether it's on (true) or off (false)
     */
    private void doFontBold( boolean boldOn )
    {
        final PlatyFont currF = gdd.getCurrFont();
        currF.lineNumber = gid.curr.lineNumber;
        currF.bold = boldOn;
```

Problems | Javadoc | Declaration | Metric - #0 - /Platypus | Audit

Resource	AC	AHF	AID	AIF	AIUR	ALD	AOFD	AUF	CBO	CC	CF	CIW	CL	CM	COC
<default>	24								25	43					
Command	27						10	0	14	43		0	94	1	0
CommandValue	4						5	0	3	7		1	100	8	0
FontSize	0				100		2	0	4	3		1	83	1	0
GDD	165						8	16	15	40		34	89	27	7
GID	36						0	100	3	2		1	100	3	4
Infile	42						4	50	12	35		6	56	2	1
Literals	18						2	75	4	3		2	57	18	10
LogFormatter	9				100		1	0	3	5		1	57	0	0
Outfile	54						11	28	25	23		6	48	4	3
ParaIndent	0				100		2	0	4	3		1	83	2	0
ParaSkip	0				100		2	0	4	3		1	83	2	0
PlatyConst	35						0	22	1	6		28	100	5	6
PlatyFont	17						5	100	6	21		1	67	6	1

Metrics in JBuilder 2007 can be shown spreadsheet style (top circle) or in a Kiviat diagram (bottom circle).

offered for years as a separate product. OptimizeIt provides numerous high-resolution views into the performance and memory consumption of the software. It includes code coverage analysis (although only as a percentage of the class covered, rather than on a line-by-line basis) and other insights into what is happening beneath the covers, including per-thread data.

JBuilder also offers impressive collaborative features. It sports a developer-oriented messaging system, which helps with code reviews as well as developer communication. It uses a peer-to-peer design that, unfortunately, works only with peers on the same network segment.

For team coordination, JBuilder provides TeamInsight, which is an easily configurable portal server that comprises key open-source tools: Subversion, Bugzilla, Continuum, and XPlanner. This portal is accessible through a Web interface or via JBuilder and includes numerous project reports and metrics. Neither of the other IDEs comes close to this level of team integration.

JBuilder feels solid throughout — a remarkable achievement given its status as a first release on Eclipse. The only bugs I ran into were frequent help icons that did not work. My complaints focus on features that are not implemented, such as the lack of visual designers for JSP or JSF (although these are coming shortly). The product also does not generate deployment files for applications using DB2, which is a curious omission. Finally, it currently ships on Windows only. Linux and Mac versions are slated for May.

At $1,999 for the edition I reviewed, JBuilder is not cheap, but it provides tremendous bang for the buck. For developers who don't need all the high-end features, there are professional and developer versions of JBuilder available for $799 and $399 respectively.

Sun NetBeans 5.5

Sun's NetBeans product is the only completely open-source product in this review, available at no cost from netbeans.org. Unlike the other packages, NetBeans requires a little assembly; you start with the core NetBeans platform, and add several "packs," depending on your needs.

Currently, Sun offers an Enterprise pack and a Visual Web pack (both used in this review), a mobility pack for J2ME programming, and a C/C++ pack. A profiler (also included in this review) is a separate pack. These packs are supersets of the common IDE plug-ins and generally provide substantial new functionality. Once I downloaded the packs, I installed them with no difficulty.

When I first examined NetBeans, several years ago, it was more of a tag-along IDE with some good features, rather than a true peer of the other Java IDE products. This is no longer the case, and NetBeans' popularity reflects this: A December 2006 survey by BZ Research shows that NetBeans enjoyed robust growth last year and is now in second place behind only Eclipse (which maintains a comfortable lead).

For enterprise computing, NetBeans provides several useful features, including support for Java EE 5 in the form of Sun's Glassfish project. The IDE has good tooling for services-based enterprise development be it SOA or just plain Web services. For example, NetBeans is the only product reviewed here with full diagramming and modeling capabilities for BPEL.

The enterprise services offerings are offset, however, by lack of support for common products. NetBeans does not support IBM's WebSphere app server and it lacks integrated support for any database other than JavaDB. The latter point needs some clarification, though: NetBeans will recognize any JDBC-accessible database, but it generates deployment files and exploits DBMS-specific features only for JavaDB.

Collaboration features are very good. NetBeans has built-in facilities

JBuilder feels solid throughout — a remarkable achievement given its status as a first release on Eclipse.

for real-time collaboration between developers, including chat and code-sharing capabilities.

Unlike these features in JBuilder, NetBeans' design is server-based. You can set up your own server for this communication, or use one provided by Sun at no charge. You simply login to Sun's server and any developers in your group are displayed along with their login status—a design that is similar to presence awareness in IM products.

As for GUI design, NetBeans bundles Matisse, which is the best GUI layout tool of its kind. As you drag and drop widgets onto panels and dialogs, they automatically arrange themselves correctly. Guidelines for optimal and alternative placements pop up during the drag and drop operations. Matisse then generates code from the design.

This tool alone makes NetBeans the IDE of choice for sites that do a lot of Swing-based interfaces, as Matisse works only with Swing. Fortunately, due to steady advances in Swing performance and look-and-feel, this is no longer the limitation it once was.

Whereas the other Java IDEs in this review all use their own proprietary formats to store project metadata, NetBeans smartly relies on Ant files to hold project configuration data (in fact, it uses Ant, the open-source Java equivalent of make, to drive builds). This has one important advantage: in teams that use multiple Java IDEs, any other IDE can load and run a NetBeans project without having to convert it manually or import it piecemeal.

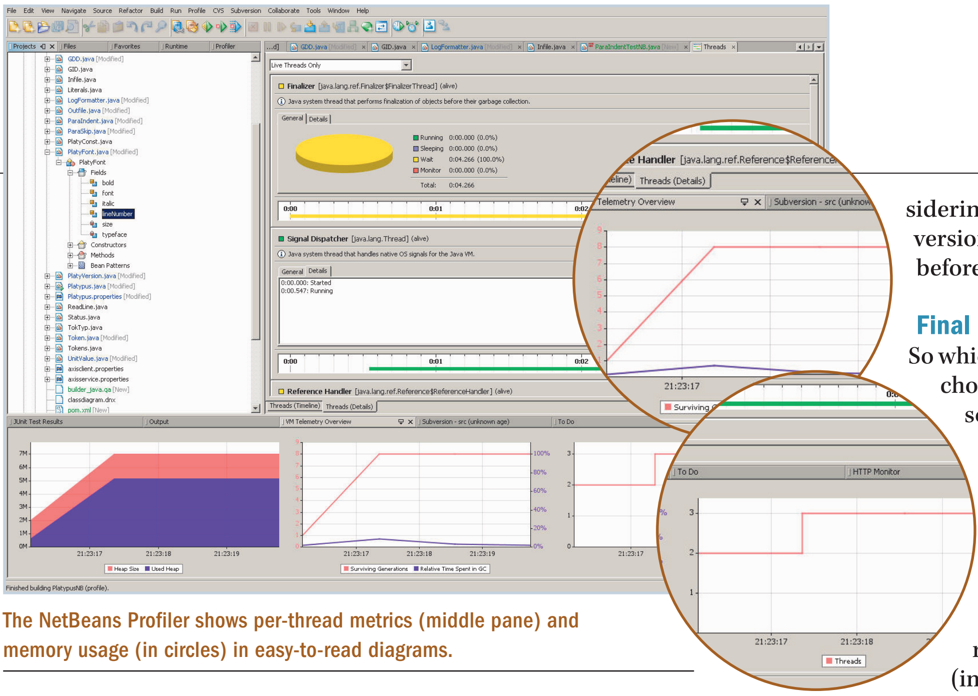
There is a downside to the use of Ant files: NetBeans supports only a single runtime configuration. Most other IDEs let you choose from as many runtime configurations as you're willing

Java Support, Coding Set IDEs Apart

Borland/CodeGear has a more complete overall feature set, but NetBeans' platform support and IBM's testing tools are areas of strength for their IDEs.

	IBM Rational Application Developer 7	Borland/CodeGear JBuilder 2007	Sun NetBeans 5.5 (Enterprise)
DIAGRAMS AND VISUAL EDITORS			
UML Diagrams	2	9	8
Code-UML diagram round-tripping	Class, sequence only	Class, sequence only	Class, sequence only
Other diagrams	Browse, Topic	EJB, Web services	BPEL
WYSIWYG visual editors	HTML, JSP, JSF	HTML	HTML, JSP, JSF[1]
GUI designers	Swing, SWT, AWT	Swing, SWT, AWT	Swing
JAVA CODING			
Support for Java SE 6	No	Complete	Minimal
Static code analysis tools	Proprietary	PMD, Findbugs, Proprietary	None
Spell checking comments and literals	Broken	No	No
Code metrics	No	Extensive	No
Fix-n-go debugging/ remote debugging	Yes/Yes	Yes/Yes	Yes/Yes
Process to generate Ant file	Complex	Trivial	Trivial
TESTING AND TUNING			
Performance/memory profilers	Yes/Yes	Yes/Yes	Yes/Yes
JUnit test generation	Stubs	No	Stubs
Web services testing tool/HTTP monitor	Yes/TCP/IP monitor	Yes/Yes	Yes/Yes
ENTERPRISE JAVA			
Support for Java EE5	None	Complete	Complete
Create files for J2EE servers	Geronimo, JBoss, WebLogic, WebSphere	Geronimo, Glassfish, JBoss, Oracle, WebLogic, WebSphere,	Glassfish, JBoss, Sun, WebLogic
DBMS built-in support	9	9 (but not DB2)	JavaDB only
Other data support	SDO	EJB3, Hibernate	JPA
MISCELLANEOUS			
Collaboration features	Minimal	Extensive	Good
Foreign languages supported	Many	Few	Few
Platforms	Windows, Linux	Windows[2]	Windows, Linux, MacOS, Solaris

[1]Requires freely available Visual Web Pack
[2]Linux and Mac versions ship in May


Finalizer [java.lang.ref.Finalizer$FinalizerThread] (alive)

Java system thread that performs finalization of objects before their garbage collection.

Signal Dispatcher [java.lang.Thread] (alive)

Java system thread that handles native OS signals for the Java VM.

Reference Handler [java.lang.ref.Reference$ReferenceHandler] (alive)

Finished building PlatypusNB (profile).

The NetBeans Profiler shows per-thread metrics (middle pane) and memory usage (in circles) in easy-to-read diagrams.

to write; not NetBeans. Instead, you must change the one configuration by hand each time you want to change the parameters you pass to your application. (The upcoming 6.0 release of the IDE remedies this problem.)

I ran into no bugs using NetBeans and it has a snappy feel except when running instrumented code in the profiler. My only complaint about the user experience is that Sun does not use anti-aliased fonts, so text is more difficult to read than in Eclipse-based solutions.

Clearly, NetBeans has an unusual mix of features—some superbly implemented, others entirely missing. If the mix of features appeals to you, NetBeans is definitely your ticket. Not only is it free, but it is snappier than the Eclipse-based products and easier to navigate, as it forgoes the "views" design embraced by Eclipse and simply uses windows. Also, NetBeans is frequently revved, enjoys a very active community, and benefits from a plug-in inventory second only to that of Eclipse.

To be fair, NetBeans is most disfavored by the timing of this review—the company is preparing version 6 of its IDE, which fixes many of my complaints including the fonts and the run-time configuration. If you're considering NetBeans, examine the version 6 beta currently available before making your decision.

Final Round-Up

So which of these IDEs should you choose? If you're running IBM's software stack or you have multiple languages spoken at your site, RAD 7 is your best bet — as long as you don't need support for Java EE 5 or Java SE 6.

If you want an inexpensive solution or one that runs on Mac OS and Solaris (in addition to Windows and Linux), your choice is NetBeans. For all other situations, JBuilder 2007 is the clear choice — and a truly standout IDE.

More Tools to Try

THE KEY TO CHOOSING AN IDE IS SELECTING ONE THAT MAKES YOU AS PRODUCtive as comfortably possible. Here's a brief overview of three alternatives to the IBM, Borland/CodeGear, and Sun NetBeans products discussed in this review.

Eclipse. If you don't need the advanced features of RAD 7 or JBuilder, you might consider a vanilla version of Eclipse, which is available at no cost from eclipse.org. If you need a few advanced features, consider MyEclipse from Genuitec (infoworld.com/4546), which integrates many open-source tools (including NetBeans' Matisse) into Eclipse for $54/year per seat.

JetBrains IntelliJ. This Java IDE is considered by many developers to be the most productive and enjoyable environment for pure coding. IntelliJ is more intuitive than the IBM, Borland, or Sun IDEs, which is how it earns its great reputation. It has some unique features, too. For example, code rules are run in background, so that errors or poor style show up as you code, and correcting the issues results in their immediate removal from the screen. These rules are more numerous than in any of the three products in this review. For sites that write a lot of Java code and do not need modeling tools, IntelliJ is a very strong, inexpensive option.

Oracle JDeveloper. JDeveloper is a feature-rich and free (but not open-source) Java IDE available at Oracle's site (infoworld.com/5147). Like NetBeans, it is not based on Eclipse. It has strong support for enterprise features including SOA and Web services and, of course, special integration with Oracle's database technologies and OC4J Java application server. *— A.B.*

Application Factories: Moving from a Generic IDE to an Application-specific IDE

Development tools have come a long way over the last few decades. Today's IDEs are an indispensable part of any developer's tool box - most tout a rich array of features that span the entire development lifecycle.

Tools vendors such as CodeGear have made significant contributions to this evolution. For example, CodeGear JBuilder® includes features such as a Visual Workbench for EJB, JPA and Web services development; TeamInsightTM to manage complex projects across multiple locations through distributed development and collaboration tools; powerful code archeology tools such as LiveSourceTM UML; code audit and metrics tools; and OptimizeITTM performance management tools for profiling, code coverage and JEE performance among others.

Now, what's next for IDEs?

Let's take a look at software development today.

Over the last few decades a ton of code has been written, and thanks to open source, a large part of it is in the public domain. The capability as well as the complexity of applications has continued to grow exponentially. Most applications are developed in teams, often geographically distributed. Most applications have evolved over a number of years with many nuances, patterns and best practices very specific to each application, and with a tremendous flux of personnel that leave their fingerprints on applications throughout their evolution.

One of the biggest challenges in application development today is to factor in application-specific information. The answer lies in the ability to capture application evolution and developers' knowledge of building and extending the application as metadata so that it's not lost in time or translation. Once that can be done effectively, the combination of the application code with metadata pertaining to application evolution can be meaningfully packaged into reusable software asset libraries.

Through what we call "Application Factories", the IDE can be transformed to play a central role in capturing both the application evolution and the developers' knowledge.

What are Application Factories?

Let's start by saying Application Factories are a set of tools to enable producing and consuming of Application Modules.

What are Application Modules?

Application Modules are simply a collection of application source artifacts as standard projects accompanied by Application Factory Metadata.

What is Application Factory Metadata?

Application Factory Metadata is a repository of application-specific artifacts. These artifacts capture the structure, evolution, and logic used in the development and maintenance of the accompanying application.

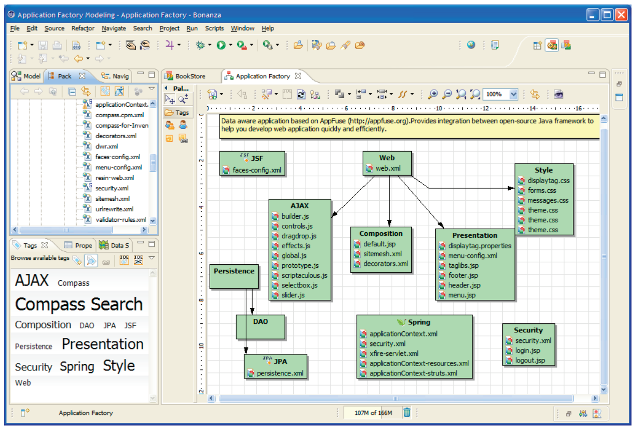
How is the metadata manifested in my day to day development?

This manifestation takes several forms: new behavior overlaid on existing views, new views, editors and context actions. The logic captured as scripts contributes new code-generation options. Code-generation in turn leverages the Application Factory framework to enable learning and archeology. All these capabilities working together transform the IDE from a generic IDE to an Application-specific IDE.

Let's illustrate this phenomenon of transforming the IDE from one that is generic to one that's application-specific.

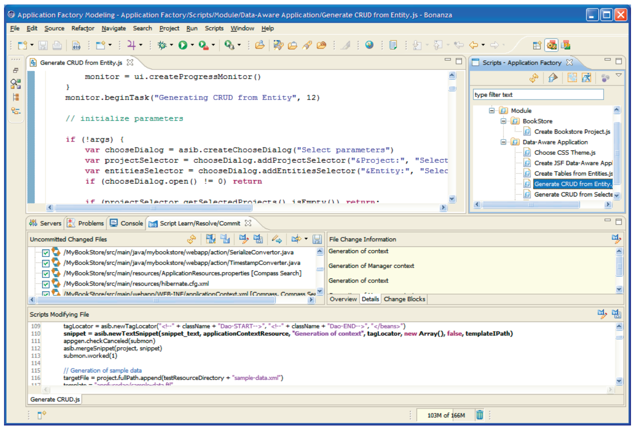
In most development tools today, you navigate the application based on a generic explorer paradigm, superimposed by a domain – say Java projects and packages. Diagramming tools navigate to resources based on a modeling construct. Quick search tools jump to known resources quickly. All of these tools are very useful, generic IDE tools which will work with any application.

Application Factories will transform these very tools and add additional ones by applying application-specific metadata. These tools not only let you navigate based on a generic construct, but let you both organize and navigate based on an application-specific construct. Some of these help in manifesting application comprehension, visualization and anchored navigation. These are some of the features that help others (and the original developer, six months later) remember and understand the nuances of the application. This is illustrated in Figure 1.


Data aware application based on AppFuse (http://appfuse.org).Provides integration between open-source Java framework to help you develop web application quickly and efficiently.

Take another example to illustrate the transformation from the generic to the application-specific using scripts. (But how did the script get created in the first place? The application module producer chose to use a script as an efficient way to deliver new functionality. The Application Factory producer tools help to create such a script. I will discuss the process of creating scripts in Part 2 of this article. For the sake of this illustration assume that a module contains the collected wisdom from people who needed to do something similar on the same type of framework)

The illustration is a Web application module which contains a script to add AJAX enabled type-ahead capability to a HTML table in your web application. Application Factories surface the presence of that kind of metadata and lets you execute it in the Application Factory framework, prompting for user input as needed. As you do so, Application Factories prepares by generating the code modifications required and presents it in an optional learn mode with various kinds of customization options. One of the primary customizations available now is to modify the code generation itself, adapting it as the application evolves over time. On acceptance of the presented modifications in the learn mode, the application is modified. This is illustrated in Figure 2.


Application Factory Modeling - Application Factory/Scripts/Module/Data-Aware Application/Generate CRUD from Entity.js - Bonanza
Generate CRUD from Entity.js
Scripts - Application Factory
Create Bookstore Project.js
Data-Aware Application
Choose CSS Theme.js
Create JSF Data-Aware App
Create Tables from Entities.js
Generate CRUD from Entity
Generate CRUD from Selecte
Script Learn/Resolve/Commit
Uncommitted Changed Files
File Change Information
Scripts Modifying File
Generate CRUD.js

```
    monitor = ui.createProgressMonitor()
}
monitor.beginTask("Generating CRUD from Entity", 12)

// initialize parameters

if (!args) {
    var chooseDialog = asib.createChooseDialog("Select parameters")
    var projectSelector = chooseDialog.addProjectSelector("&Project:", "Select
    var entitiesSelector = chooseDialog.addEntitiesSelector("&Entity:", "Selec
    if (chooseDialog.open() != 0) return
```

```
/MyBookStore/src/main/java/mybookstore/webapp/action/SerializeConvertor.java
/MyBookStore/src/main/java/mybookstore/webapp/action/TimestampConverter.java
/MyBookStore/src/main/resources/ApplicationResources.properties [Compass Search]
/MyBookStore/src/main/resources/hibernate.cfg.xml
```

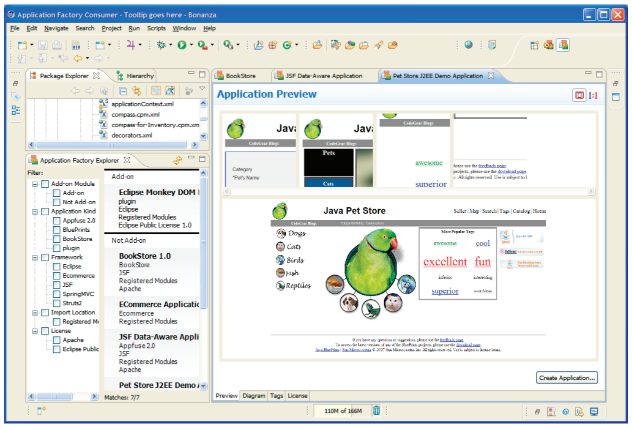
```
Generation of context
Generation of Manager context
Generation of context
```

```
        tagLocator = asib.newTagLocator("<!--" + className + "Dao-START-->", "<!--" + className + "Dao-END-->", "</beans>")
        snippet = asib.newTextSnippet(snippet_text, applicationContextResource, "Generation of context", tagLocator, new Array(), false, templateIPath)
        appgen.checkCanceled(submon)
        asib.mergeSnippet(project, snippet)
        submon.worked(1)

        // Generation of sample data
        targetFile = project.fullPath.append(testResourceDirectory + "sample-data.xml")
```

The examples above help illustrate how Application Factories with the help of metadata transform the IDE from a generic tool to an application-specific tool. But, how do the Application Module and the accompanying metadata get created in the first place? This is where the consumer part of Application Factories comes into play.

Using the Application Factories tools you can browse pre-created Modules. The browsing of the module is again anchored on metadata – allowing you to filter based on arbitrary facets, resulting look and feel of the application, the architectural view of the application's internals and others. A view of the module browser is shown in Figure 3.



An Application-first Development Model

The approach of creating application metadata to capture the intent and knowledge of developers provides an application-driven development model. The IDE now evolves to take the role of manifesting the application-specific metadata with interactive, just-in-time tooling.

This approach to creating metadata that easily captures the intent and knowledge of developers working on an application provides a new model for designing software. Applications evolve over a number of years with many nuances, patterns and best practices very specific to each application. Many developers work on an application throughout its lifecycle, gaining knowledge of its inner workings and developing effective ways to maintain and enhance it. These are invaluable assets to hold on to and deploy as modules. Starting from such a base, developers can customize and build applications more efficiently—thereby greatly reducing the cost and overhead associated with the complexity of today's applications.

Such a model addresses the core challenges in software development today – avoiding constant relearn, capturing valuable data about the application, application reuse, and a meaningful software asset library.

For more information on Application Factories watch for upcoming announcements at www.codegear.com and read about CodeGear's Eclipse-based JBuilder, JGear and 3rdRail IDEs. We invite you as well to join in discussions and share best practices with the more than 6 million developers doing Java, Eclipse, Rails, Windows, PHP and database development in our CodeGear's Developer Network - http://dn.codegear.com/
— *Ravi Kumar*

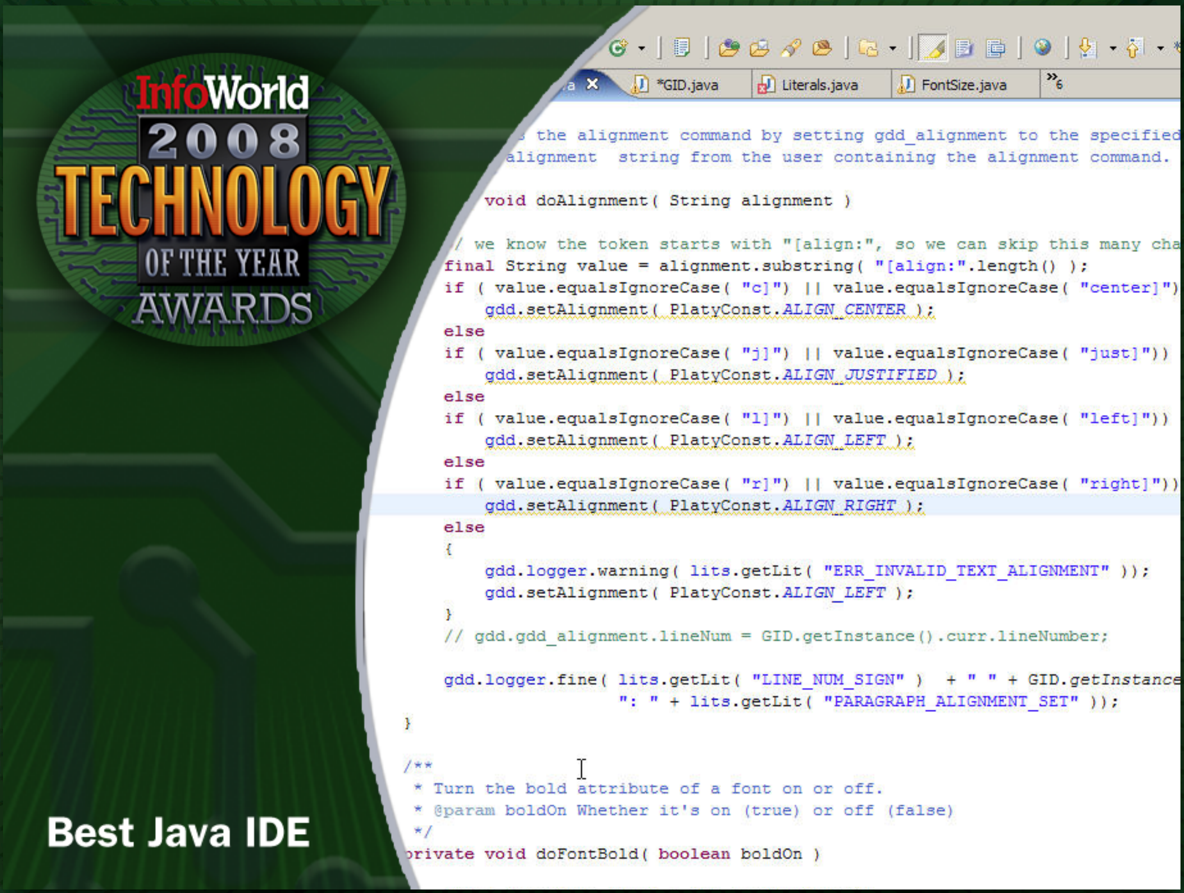
Ravi Kumar is principal architect in the Java tools group at CodeGear and is responsible for the vision and architecture of CodeGear's Eclipse-based JBuilder product line. He was the driving force behind the JBuilder ProjectAssist team system vision. He specializes in SOA, Web Services and Database tooling. These days he is working on making the new paradigm of Application Factories a reality in future versions of CodeGear products.

CONGRATULATIONS

CodeGear

for being named

BEST JAVA IDE



```
 s the alignment command by setting gdd_alignment to the specified
 alignment  string from the user containing the alignment command.

  void doAlignment( String alignment )

 / we know the token starts with "[align:", so we can skip this many cha
final String value = alignment.substring( "[align:".length() );
if ( value.equalsIgnoreCase( "c]") || value.equalsIgnoreCase( "center]")
    gdd.setAlignment( PlatyConst.ALIGN_CENTER );
else
if ( value.equalsIgnoreCase( "j]") || value.equalsIgnoreCase( "just]"))
    gdd.setAlignment( PlatyConst.ALIGN_JUSTIFIED );
else
if ( value.equalsIgnoreCase( "l]") || value.equalsIgnoreCase( "left]"))
    gdd.setAlignment( PlatyConst.ALIGN_LEFT );
else
if ( value.equalsIgnoreCase( "r]") || value.equalsIgnoreCase( "right]"))
    gdd.setAlignment( PlatyConst.ALIGN_RIGHT );
else
{
    gdd.logger.warning( lits.getLit( "ERR_INVALID_TEXT_ALIGNMENT" ));
    gdd.setAlignment( PlatyConst.ALIGN_LEFT );
}
// gdd.gdd_alignment.lineNum = GID.getInstance().curr.lineNumber;

gdd.logger.fine( lits.getLit( "LINE_NUM_SIGN" )  + " " + GID.getInstance
                 ": " + lits.getLit( "PARAGRAPH_ALIGNMENT_SET" ));
}

/**
 * Turn the bold attribute of a font on or off.
 * @param boldOn Whether it's on (true) or off (false)
 */
rivate void doFontBold( boolean boldOn )
```

CODE GEAR™ FROM Borland®

InfoWorld