# CODEGEAR FROM Borland — JBUILDER 2008 Feature Matrix

## Certified Eclipse Bundle

| | Turbo | Professional | Enterprise |
|---|---|---|---|
| Integrated Installation | >> | >> | >> |
| Sample Applications | >> | >> | >> |
| Eclipse 3.3.2 (Europa) Framework with WTP 2.0, including: | >> | >> | >> |
| Business Intelligence and Reporting Tools (BIRT) | >> | >> | >> |
| CVS plug-in | >> | >> | >> |
| Dali Java Persistance Tools (part of WTP) | >> | >> | >> |
| Data Tools Platform (DTP) | >> | >> | >> |
| Eclipse Modeling Framework (EMF) | >> | >> | >> |
| Graphical Editing Framework (GEF) | >> | >> | >> |
| J2EE Standard Tools (JST) | >> | >> | >> |
| Java Development Tools (JDT) | >> | >> | >> |
| Mylyn (formerly Mylar) | >> | >> | >> |
| Plug-in Development Environment (PDE) | >> | >> | >> |
| Rich Client Platform (RCP) | >> | >> | >> |
| Test and Performance Tools Platform (TPTP) | >> | >> | >> |
| Visual Editor (VE) | >> | >> | >> |
| Web Standard Tools (WST) | >> | >> | >> |
| Additional Eclipse and Third-Party Plug-Ins (partial list): | | | |
| Apache | >> | >> | >> |
| Apache Ant | >> | >> | >> |
| Apache Maven 2.0 Integration | >> | >> | >> |
| Apache Xerces | >> | >> | >> |
| Eclipse AspectJ Development Tools (AJDT) | >> | >> | >> |
| Eclipse Communcation Framework (ECF) | >> | >> | >> |
| Eclipse Graphical Modleing Framework (GMF) | >> | >> | >> |
| Eclipse JavaServer Face Tools (JSF) | >> | >> | >> |
| Eclipse Modeling Framework Technology (EMFT) | >> | >> | >> |
| Eclipse UML2 | >> | >> | >> |
| Eclipse XSD (part of Eclipse MDT) | >> | >> | >> |
| Hibernate | >> | >> | >> |
| JUnit | >> | >> | >> |
| mySQL | >> | >> | >> |
| Spring IDE | >> | >> | >> |

## Java EE Support

| | | | |
|---|---|---|---|
| LiveSource graphical view of EJB 2.1 projects | | >> | >> |
| LiveSource graphical view of EJB 3.0 projects | | >> | >> |
| LiveSource graphical view of JPA (Hibernate/TopLink) projects | | >> | >> |

# JBUILDER Feature Matrix

| | | |
|---|---|---|
| Visual point-and-click two-way designer of entity, session, and message beans. Two-way generation of classes and annotations | >> | >> |
| Visually design CMP 2.x relationships and configure database mapping | >> | >> |
| Visually design EJB 3.0/JPA entity relationships and configure database mapping | >> | >> |
| Configure XDoclet task properties for server runtimes supported in JBuilder | >> | >> |
| Visually configure persistence properties for EJB 3.0/JPA | >> | >> |
| Full build/deploy/redeploy capabilities | >> | >> |
| Visually create session beans | >> | >> |
| Automatically arrange beans on design panes | >> | >> |
| Create multiple design panes to logically group beans in an EJB group | >> | >> |
| Adjust filtering to hide or show generated classes | >> | >> |
| Import EJB descriptors, including vendor-specific descriptors, for all supported servers | >> | >> |
| Visually create message-driven beans | >> | >> |
| Visually create entity beans | >> | >> |
| Convert descriptor-based source to XDoclet or Java EE 5 annotations | >> | >> |
| Import multi-module Java EE projects from earlier versions of JBuilder and convert them to native Eclipse projects | >> | >> |
| Generate EJB 2.x entity beans from existing database schema | >> | >> |
| Generate EJB 3.x entity beans from existing database schema | >> | >> |
| Instant navigation between visual EJB components in designer and source | >> | >> |
| Visually create OCL constraints | >> | >> |
| Fully integrated with Eclipse 3.3 framework and Eclipse WTP 2.0 | >> | >> |
| VisiBroker builder - integration to VisiBroker tools | >> | >> |
| Visual representation of Java/EJB with methods and operations | >> | >> |

## APPLICATION SERVERS SUPPORTED

| | | | |
|---|---|---|---|
| Apache Geronimo 1.1.1 | >> | >> | >> |
| Apache Geronimo 2.0 | >> | >> | >> |
| Apache Tomcat 6.0 | >> | >> | >> |
| BEA WebLogic Application Server 10.0 | >> | >> | >> |
| BEA WebLogic Application Server 8.1 | >> | >> | >> |
| BEA WebLogic Application Server 9.2 | >> | >> | >> |
| Borland Enterprise Server 6.7 | >> | >> | >> |
| IBM WebSphere 6.0 | >> | >> | >> |
| IBM WebSphere 6.1 (with EJB 3 feature pack) | >> | >> | >> |
| JBoss 3.2.3 | >> | >> | >> |
| JBoss 4.0.5 | >> | >> | >> |
| JBoss 4.2.2 | >> | >> | >> |
| Oracle Application Server 10.1.3.2 | >> | >> | >> |
| Oracle Containers for Java (OC4J) 10.1.3.2 | >> | >> | >> |
| Sun Glassfish 1.0 | >> | >> | >> |
| Sun Glassfish 1.1 | >> | >> | >> |

| | | | |
|---|---|---|---|
| Sun Glassfish 2.0 | >> | >> | >> |

## Web Services Support

### WEB SERVICES DESIGNER

| | | |
|---|---|---|
| Visual two-way designer for creating Axis based Web Service-enabled applications | >> | >> |
| Streamlined user interface and wizards | >> | >> |
| Visual representation of WSDL with methods and operations | >> | >> |
| Activate or deactivate services without removing components from the designer | >> | >> |
| Instant navigation between visual Web Services components in designer and source | >> | >> |
| Import Web services projects from earlier versions of JBuilder, and convert them to native Eclipse projects | >> | >> |
| Deploy an EJB as a Web Service | >> | >> |

### BUILD, DEBUG, AND RUN WEB SERVICES

| | | |
|---|---|---|
| Create an Axis deployment environment | >> | >> |
| Dynamic generation and deployment of Web Services components during project builds | >> | >> |
| Deploy Axis runtime to Tomcat, JBoss, WebLogic, WebSphere, Geronimo, Glassfish, Oracle | >> | >> |

## Collaboration and Team Development with ProjectAssist and TeamInsight

### CREATE NEW TEAM PROJECT ENVIRONMENT DEFINITION

| | | | |
|---|---|---|---|
| Wizard to create a new deployment model | | | >> |
| Options for deep and shallow scanning for pre-existing services | | | >> |
| Create new installation or assimilate existing installation for source code management | | | >> |
| Create new installation or assimilate existing installation for defect tracking | | | >> |
| Create new installation or assimilate existing installation for requirements management and project planning | | | >> |
| Create new installation or assimilate existing installation for continuous integration builds | | | >> |
| Create new projects on a new or existing server deployment, across all services | | | >> |
| Create new users, add users to new or existing projects | | | >> |
| Mylar (OSS) | >> | >> | >> |
| XPlanner Mylar Connector | | >> | >> |
| StarTeam Mylar Connector | | | >> |

### INSTALL NEW TEAMINSIGHT STACK

| | |
|---|---|
| Deep or shallow scan of target server for potential conflicts | >> |
| Full deployment and automated configuration of all services | >> |

| | | | |
|---|---|---|---|
| Automated test of deployment to verify the system is operational | | | >> |

### SUPPORTED TOOLS FOR TEAMINSIGHT

| | | | |
|---|---|---|---|
| Bugzilla | | | >> |
| Continuum | | | >> |
| CVS | | | >> |
| StarTeam | | | >> |
| Subversion | | | >> |
| XPlanner | | | >> |

### PROJECT PORTAL

| | | | |
|---|---|---|---|
| Monitor activity in source code repository for project, track recent check-ins | | | >> |
| Monitor quality metrics including tables of bugs by severity, by product area, by owner, newest bugs, and bug find/fix rates | | | >> |
| Monitor team velocity via live burn-down charts | | | >> |
| Monitor team progress against committed features, feature-by-feature | | | >> |
| Monitor continuous integration builds, track failed builds to identify root-cause | | | >> |

### IDE INTEGRATION

| | | | |
|---|---|---|---|
| Automatic configuration of IDE to pull project from CVS | | | >> |
| Automatic configuration of IDE to pull project from Subversion | | | >> |
| Single-pane view of individual's project responsibilities: Assigned Tasks, Requirements Owned, Requirements Tracked, Assigned Bugs, Reported Bugs, and Code To-Do's | | | >> |
| Project portal and all services available via integrated, tabbed browser in IDE | | | >> |
| Integrated creation and editing of bug reports | | | >> |
| Integrated creation and editing of requirements and tasks | | | >> |
| Full off-line mode for persistent access to requirements, tasks, defects with automated synchronization | | | >> |
| Serverless LAN peer discovery and chat | >> | >> | >> |
| Peer code reviews | | | >> |
| Open API to enable integration with other popular version control systems such as Perforce, ClearCase, and ClearQuest | | >> | >> |

## Additional Productivity Features

| | | | |
|---|---|---|---|
| RMI builder | | >> | >> |
| JNI builder | | >> | >> |
| Import JBuilder 2006 projects | >> | >> | >> |
| Quality Assurance: Code audits and metrics | | | >> |
| Design Patterns support, including GoF patterns with code template design and re-use | | >> | >> |

# JBUILDER Feature Matrix

## Code Profiling and Performance Tuning Powered by Optimizeit™ Technology

| | | |
|---|---|---|
| Seamless integration with JBuilder | >> | >> |
| Complete code profiling and performance management capabilities for identifying and solving code-level performance issues. | >> | >> |
| Improve performance and reliability of any Java code: Java applications, Java EE applications, servlets, applets, EJBs, JavaBeans, JSP applications, and Java Tag Libraries | >> | >> |
| Easily connects to a remote Java process to test a program running on a different machine or a program launched outside of JBuilder | >> | >> |

### MEMORY AND CPU PROFILING POWERED BY OPTIMIZEIT

| | | |
|---|---|---|
| High-level performance-related data displayed in real time in order to determine whether a performance issue is related to CPU, memory, or both | >> | >> |
| Automatic Memory Leak Detector monitors the evolution of memory usage over time for the immediate identification of potential memory leaks | >> | >> |
| Real-time monitoring of object allocations to understand how the profiled program uses the virtual machine memory; Allocation Backtrace View enables identification of the code or part of the program responsible for object allocations | >> | >> |
| Object Size Display automatically computes and displays, in real time, the amount of memory being consumed by all instances of a class; then sort and view by object size to prioritize objects consuming the most memory | >> | >> |
| Reduced reference graph provides a transitive closure of the full reference graph to display only references that should be removed in order to free the object for garbage collection | >> | >> |
| CPU Profiler measures pure CPU usage or time usage during a profiling session, with option to use sampling-based or instrumentation-based profiler | >> | >> |
| Display profiling information per thread and thread groups, with color highlighting of threads that were busy during profiling session | >> | >> |
| HotSpot Display lists methods where most time was spent, to help identify bottlenecks due to single methods | >> | >> |
| Scalable call graph visually isolates critical code. Select a string allocation and highlight the flow of a method call to see where memory and time are being spent | >> | >> |
| Automatic Application Quality Analyzer supports performance-error prevention and coding standards by automatically detecting VM-level performance bugs | >> | >> |
| Export views in XML, HTML and CSV format | >> | >> |

### THREAD DEBUGGER POWERED BY OPTIMIZEIT

| | | |
|---|---|---|
| Real-time display of the progress of all threads running within the virtual machine | >> | >> |
| Understand thread contentions for a monitor with the detailed panels of the Contention View | >> | >> |
| Wait state monitoring to understand why a thread is not making progress with the Waiting View and I/O Waiting View | >> | >> |

| | | |
|---|---|---|
| Identify and correct excessive locking where a thread enters and holds monitors using the Monitor Enter View reports | >> | >> |
| Analyze deadlocks using graphical view of the relationships between threads and monitors to quickly understand deadlock situations | >> | >> |
| Predict deadlocks with Monitor Usage Analyzer which generates full list of warnings and errors that might lead to deadlocks and performance bottlenecks, such as lock order warnings, lock and wait warnings, and lock and I/O wait warnings | >> | >> |

### CODE COVERAGE POWERED BY OPTIMIZEIT

| | | |
|---|---|---|
| Real-time Class Coverage View to quickly see the coverage for each class and identify classes not fully covered | >> | >> |
| Real-time display of all classes and interfaces used by the tested program and real-time percentage of lines covered per class | >> | >> |
| Method Coverage View displays the methods and lines of code for a selected class that have not been used, allowing developers to modify test plans to cover all areas of the code | >> | >> |
| Source Code Viewer shows lines of code that have never been executed, making it easier to spot dead code | >> | >> |
| Option to display the interfaces that have been loaded by the virtual machine and those that have not been loaded | >> | >> |
| Batch-mode support to easily include code coverage in any batch-mode testing process | >> | >> |

### REQUEST ANALYZER POWERED BY OPTIMIZEIT

| | | |
|---|---|---|
| Profile the performance behavior of your Java EE application code across the following Java EE components: JDBC, JSP, JNDI, Enterprise JavaBeans, and JMS containers | >> | >> |
| Improve performance and reliability of Java EE-related application code earlier in development with drill-down performance information for Java EE components | >> | >> |
| Visual interface simplifies the complexity of Java EE application interactions using graphical representation | >> | >> |
| System Dashboard view provides a graphical display of the application time spent in Java EE components and total number of requests. Shows the percentage of use for each server module to quickly detect any major component-level performance issues | >> | >> |
| System Composite view displays all of the Java EE events that have occurred in an application, in real time, in their proper hierarchy. Hierarchy shows the relationship of events in terms of which events spawn others | >> | >> |

## UML Modeling and Code Archeology Powered By Together® Technology

| | | |
|---|---|---|
| Generate HTML portal documentation | >> | >> |
| Generate documentation using template | >> | >> |
| Generate image files from UML diagrams | >> | >> |
| Automatic generation of sequence diagrams | >> | >> |
| Use Case diagrams | | >> |
| Activity diagrams | >> | >> |

| | | |
|---|---|---|
| Sequence diagrams | >> | >> |
| Class diagrams | >> | >> |
| Component diagrams | | >> |
| Deployment diagrams | | >> |
| State Machine diagrams | | >> |
| Composite Structure diagrams | | >> |
| Communication diagrams | | >> |
| Web services diagrams | >> | >> |

## Developer Productivity and Code Reuse with JBuilder Application Factories

| | |
|---|---|
| Ability to organize code visually. track changes. associate changes to actions. data mine actions from the past, and associate all project artifacts in the context of the desired user story or task. | >> |
| Application Module Editor | >> |
| Application Module Runtime | >> |
| Applicaton Factory Explorer | >> |
| Recipe Editor | >> |
| Data-aware Web application modules for Struts 2, Spring MVC, JSF | >> |
| E-commerce application module | >> |
| Template application modules for PetStore and Book Store | >> |
| Support for 3 user roles: Consumer Role, Producer Role, and Ad-hoc Developer Role | >> |
| Code generation/templating mechanisms that provide a way to generate template code via Application Module Scripts | >> |

## JBuilder Swing Designer

### CODE GENERATION OPTIONS

| | | | |
|---|---|---|---|
| Bi-Directional Code Generation | >> | >> | >> |
| Read & write almost any style | >> | >> | >> |
| Block mode (default) | >> | >> | >> |
| Flat mode | >> | >> | >> |
| Prefix component creation | | >> | >> |
| Share variables | | >> | >> |
| Control variable declaration | | >> | >> |
| Create event handler stubs | | >> | >> |
| Define default variable names | | >> | >> |
| Make selected widgets fields by default | | >> | >> |
| Make all widgets fields by default | | >> | >> |
| Event handlers as anonymous classes | | >> | >> |
| Event handlers as inner classes | | >> | >> |
| Implement listener interface in parent class | | >> | >> |

# JBUILDER Feature Matrix

## SPECIAL FEATURES

| | | | |
|---|---|---|---|
| Reverse engineer hand-written code | >> | >> | >> |
| Supports user code refactoring | >> | >> | >> |
| Free-form code editing | >> | >> | >> |
| Visual Inheritance | | >> | >> |
| Custom widget support | | >> | >> |
| Palette Manager | | >> | >> |
| Embedded Composite support | | >> | >> |
| Custom Widget Templates | | >> | >> |
| Widget Morphing | | >> | >> |
| Layout Assistant | | >> | >> |
| Graphical Tab Order Editing | | >> | >> |
| Multi-Widget Select & Edit | | >> | >> |
| Marquee Select | | >> | >> |
| Align multiple widgets | | >> | >> |
| Replicate width and height | | >> | >> |
| Gradient Editor for CLabels | | >> | >> |
| Unlimited Undo/Redo | >> | >> | >> |
| Cut/Copy/Paste | >> | >> | >> |
| Sticky Mode for multi-widget add | >> | >> | >> |
| Show externalized strings | >> | >> | >> |
| Display labels in multiple locales | | >> | >> |
| Internationalization tools & wizards | | >> | >> |
| Auto-update resource bundles | | >> | >> |
| Control editor layout | >> | >> | >> |

## SWT SPEACIAL FEATURES

| | | | |
|---|---|---|---|
| SWT_AWT | | >> | >> |
| Extract Composite Refactoring | | >> | >> |
| Gradient Editor for CLabels | | >> | >> |
| Draft Mode | >> | >> | >> |

## SWING WIZARDS

| | | | |
|---|---|---|---|
| Swing JFrame | >> | >> | >> |
| Swing JDialog | | >> | >> |
| Swing JPanel | | >> | >> |
| Swing JApplet | | >> | >> |
| Swing JInternalFrame | | >> | >> |
| Swing Application | | >> | >> |

CODE GEAR FROM Borland JBUILDER Feature Matrix

## SWING CONTAINERS

| | | | |
|---|---|---|---|
| JPanel | >> | >> | >> |
| JScrollPane | >> | >> | >> |
| JSpiltPane | >> | >> | >> |
| JTabbedPane | >> | >> | >> |
| JToolBar | >> | >> | >> |
| JDesktopPane | | >> | >> |
| JInternalFrame | | >> | >> |

## SWING LAYOUTS

| | | | |
|---|---|---|---|
| Graphical Layout Feedback | >> | >> | >> |
| Null/Absolute | >> | >> | >> |
| FlowLayout | >> | >> | >> |
| BorderLayout | >> | >> | >> |
| GridLayout | >> | >> | >> |
| CardLayout | | >> | >> |
| GridBagLayout | | >> | >> |
| BoxLayout / Struts & Glue | | >> | >> |
| SpringLayout | | >> | >> |
| JGoodies FormLayout | | >> | >> |
| Java 6 GroupLayout | | >> | >> |

## SWING CONTROLS

| | | | |
|---|---|---|---|
| Support for custom / 3rd party widgets | | >> | >> |
| JButton | >> | >> | >> |
| JCheckBox | >> | >> | >> |
| JRadioButton | >> | >> | >> |
| JToggleButton | >> | >> | >> |
| JLabel | >> | >> | >> |
| JTextField | >> | >> | >> |
| JPasswordField | >> | >> | >> |
| JTestArea | >> | >> | >> |
| JFormattedTextField | | >> | >> |
| JSlider | | >> | >> |
| JScrollBar | | >> | >> |
| JList | >> | >> | >> |
| JComboBox | >> | >> | >> |
| JSpinner | | >> | >> |
| JTree | >> | >> | >> |
| JTable | >> | >> | >> |
| JToolBar.Separator | >> | >> | >> |

| | | | |
|---|---|---|---|
| JSeparator | | >> | >> |
| JProgessBar | | >> | >> |
| JOptionPane | | >> | >> |
| JTestPane | | >> | >> |
| JEditorPane | | >> | >> |
| Action | | >> | >> |
| ButtonGroup | | >> | >> |

**SWING MENUS**

| | | | |
|---|---|---|---|
| Graphical Menu Editing | | >> | >> |
| JMenuBar | | >> | >> |
| JMenu | | >> | >> |
| JPopupMenu | | >> | >> |
| JMenuItem | | >> | >> |
| JCheckBoxMenuItem | | >> | >> |
| JRadioButtonMenuItem | | >> | >> |
| Menu Separator | | >> | >> |

**AWT WIDGETS**

| | | | |
|---|---|---|---|
| Panel | | >> | >> |
| ScrollPane | | >> | >> |
| Button | | >> | >> |
| Label | | >> | >> |
| Checkbox | | >> | >> |
| Choice | | >> | >> |
| List | | >> | >> |
| Scrollbar | | >> | >> |
| TextField | | >> | >> |
| TextArea | | >> | >> |

**SWING LOOK & FEELS**

| | | | |
|---|---|---|---|
| Windows | >> | >> | >> |
| CDE/Motif | >> | >> | >> |
| Metal | >> | >> | >> |
| JGoodies Plastic | | >> | >> |
| JGoodies Plastic 3D | | >> | >> |
| JGoodies Plastic XP | | >> | >> |
| JGoodies Windows | | >> | >> |
| Liquid | | >> | >> |
| Kunststoff | | >> | >> |

# JBUILDER Feature Matrix