

Feature Matrix

JEE	DEV	PRO	ENT
LiveSource® graphical view or EJB 2.1 projects			✓
LiveSource graphical view or EJB 3.0 projects			✓
LiveSource graphical view or JPA (Hibernate/Kodo) projects			✓
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 CMP 3.x 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			✓
Create multiple design panes to logically group beans in an EJB group			✓
Adjust filtering to hide or show generated classes			✓
Import J2EE™ projects from previous versions of JBuilder			✓
Import EJB™ source code without deployment descriptors			✓
Import EJB descriptors, including vendor-specific descriptors, for all supported servers			✓
Visually create message-driven beans			✓
Visually create session beans			✓
Convert descriptor-based source to XDoclet or Java EE 5.0 annotations		✓	✓
Import multi-module Java™ EE projects from earlier versions of JBuilder, converting 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 relationships			✓
Visually create OCL constraints			✓
Fully integrated with Eclipse WTP			✓
VisiBroker® builder - integration to VisiBroker tools			✓
Visual representation of Java/EJB with methods and operations			✓
WEB SERVICES	DEV	PRO	ENT
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, converting 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 SM , Oracle®		✓	✓
PROJECT ASSIST	DEV	PRO	ENT
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 to a new or existing server deployment, across all services			✓
Create new users, add users to new or existing projects			✓
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			
Subversion			✓
Bugzilla			✓
Continuum			✓
XPlanner			✓
TEAM INSIGHT	DEV	PRO	ENT
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 Subversion			✓
Single-pane view of individual's project responsibilities: Assigned Tasks, Requirements owned, Requirements tracked, Assigned Bugs, Reported Bugs, and Code To-Dos			✓
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			✓
Live links to all processes services for active project within IDE			✓
ADDITIONAL PRODUCTIVITY FEATURES	DEV	PRO	ENT
RMI builder	✓	✓	✓
JNI builder	✓	✓	✓
Import standard JBuilder projects	✓	✓	✓
Quality Assurance: Code audits and metrics		✓	✓
Design Patterns support, including GoF patterns with code template design and re-use	✓	✓	✓

COMPLETE PERFORMANCE TUNING WITH BORLAND OPTIMIZEIT™ 2007 FOR ECLIPSE

	DEV	PRO	ENT
Seamless integration with JBuilder enables developers to run Optimizeit™ right from within the JBuilder UI	✓	✓	✓
Borland Optimizeit delivers complete performance management capabilities for identifying and solving code-level performance issues; this integrated solution comprises Borland Optimizeit Profiler, Borland Optimizeit Thread Debugger, and Code Coverage	✓	✓	✓
Improve performance and reliability of any Java code: Java applications, J2EE applications, servlets, applets, EJBs, JavaBeans™, JSP™ applications, and Java tag Libraries	✓	✓	✓
Requires no code modifications. Runs with Sun JDK™'s for Java 5 and Java 6	✓	✓	✓
Easily connects to a remote Java process to test a program running on a different machine			✓
Optimizeit Progress Tracker analyzes the performance impact of code changes by comparing two or more snapshots			✓
Graph view in Optimizeit Progress Tracker plots performance variations over time			✓
OPTIMIZEIT PROFILER FOR MEMORY AND CPU PROFILING			
High-level performance-related data displayed in real time allows developers to understand 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 allows developers to identify the code or part of the program responsible for instance allocations	✓	✓	✓
Object Size Display automatically computes and displays, in real time, the amount of memory being consumed by all instances of a class; view and sort 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: Measure pure CPU usage or time usage during a profiling session, with option to use sampling-based or instrumentation-based profiler	✓	✓	✓
Displays 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	✓	✓	✓
New 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	✓	✓	✓
PDF Reporting: Generate reports in PDF to capture performance data	✓	✓	✓
OPTIMIZEIT THREAD DEBUGGER			
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: understand why a thread is not making progress with the Waiting View and I/O Waiting View		✓	✓
Track excessive locking: Monitor Enter View reports where a thread enters and holds monitors, allowing developers to understand and correct excessive locking		✓	✓
Analyze Deadlocks: graphical view of the relationships between threads and monitors to quickly understand deadlock situations		✓	✓
Predict deadlocks: Monitor Usage Analyzer 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		✓	✓
OPTIMIZEIT CODE COVERAGE			
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	✓	✓
SnapshotMerger merges different runs of the same application from the command line to see coverage of the combined runs	✓	✓
Generate customized, detailed coverage reports directly from the command line	✓	✓
OPTIMIZEIT REQUEST ANALYZER		
Profile the performance behavior of your J2EE application code across the following J2EE components: JDBC™, JSP, JNDI, Enterprise JavaBeans, and JMS containers		✓
Improve performance and reliability of J2EE-related application code earlier in development with drill-down performance information for J2EE components		✓
Visual interface simplifies the complexity of J2EE application interactions using graphical representation		✓
System Dashboard view provides a graphical display of the application time spent in J2EE 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 J2EE 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

	DEV	PRO	ENT
UML modeling with Borland Together® LiveSource for Java	✓	✓	✓
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		✓	✓

About Borland

Founded in 1983, Borland Software Corporation (NASDAQ: BORL) is the global leader in platform independent solutions for Software Delivery Optimization. The company provides the software and services that align the people, process, and technology required to maximize the business value of software. To learn more about delivering quality software, on time and within budget, visit <http://www.borland.com>.

Borland, LiveSource, Together, JBuilder, VisiBroker, Optimizeit and all other Borland brand and product names are service marks, trademarks or registered trademarks of Borland Software Corporation or its subsidiaries in the United States and other countries. J2EE EJB Java, JavaBeans, JSP, JDK, JVM, J2SE Java HotSpot, JDBC and all other Sun brand and product names are service marks, trademarks or registered trademarks of Sun Microsystems, Inc. or its subsidiaries in the United States and other countries. IBM, WebSphere and all other IBM brand and product names are service marks, trademarks or registered trademarks of International Business Machines Corporation or its subsidiaries in the United States and other countries. Oracle and all other Oracle brand and product names are service marks, trademarks or registered trademarks of Oracle Corporation or its subsidiaries in the United States and other countries. All other marks are the property of their respective owners.