

NetBeans™ IDE 4.1



Enabling and Accelerating Development of Java™ software for Desktop, Web Tier, Web Services, J2EE™ and Mobile Application Development

Highlights

- Create mobile, desktop, Web, and enterprise applications with a single tool.
- Superior out-of-box experience — no hunting for plug-ins.
- 100% Pure Java™, open source, free, supports Java™ standards.
- Simplifies support for J2EE™ 1.4 including Enterprise JavaBeans™ and Web services. Developers can target Tomcat, Sun Java System Application Server, JBoss, WebSphere, and WebLogic.
- Best mobility support: Only IDE with an easy-to-use solution to the device fragmentation problem.
- Project System: The first IDE to base its project system entirely on Apache Ant.



The full-featured NetBeans Integrated Development Environment (IDE) is an open source, multiplatform, zero cost, 100% Pure Java™ IDE that now includes support for Java™ 2 Platform, Enterprise Edition (J2EE™) 1.4 technology. This enables you to build rich client, Web tier, Web service, mobile, and J2EE applications. With an Ant-based project management system and a performance profiler, find out what more and more users already know — NetBeans software is the one IDE you need to develop Java applications.

NetBeans IDE is available at www.netbeans.org as a free download for the platforms developers need most — Microsoft Windows, Linux, the Solaris™ Operating System (OS), and Mac OS X. Since the IDE itself is written in the Java language, you can run it on any operating system for which a Java 2 Platform, Standard Edition (J2SE™) Java Developer Kit (JDK™) version 1.4.2, version 5.0, or later is available.

NetBeans 4.1 is more than just a development platform. It delivers tightly integrated deployment and debugging of Web and enterprise applications on Apache Tomcat 5 and the Java System Application Server 8.1 (which is the J2EE 1.4 Reference Implementation), improving developer productivity through the optimization of the develop, test, and deploy cycle. Additional support is included for JBoss, WebSphere, and WebLogic. NetBeans IDE also offers a powerful GUI builder that allows for rapid creation of rich clients using the Java Foundation Classes (JFC/Swing).

What makes NetBeans IDE special

When you use the NetBeans IDE, you enjoy the benefits of a top-shelf IDE, without the negatives associated with moving development to a single environment.

Unlike other IDEs, NetBeans IDE does not force you to use a build structure with project metadata that must be reverse engineered if you want to build the project outside of the IDE. Instead, NetBeans IDE builds on existing open standards to help automate the development process without locking you in. NetBeans IDE bases its project system on Apache Ant, the de facto standard build tool for Java applications. The project metadata produced by the NetBeans IDE is in the form of XML and properties files that can be used by Ant outside of the IDE. This means that developing a project in NetBeans IDE does not lock you into NetBeans IDE.

You can use NetBeans IDE to create large projects with sophisticated build parameters. Where such projects are already in place, you can adapt NetBeans IDE to work with them without necessarily changing the project's structure. If you are generally more comfortable with command line tools because of their transparency and the level of control they allow over your projects, NetBeans IDE could become the first IDE that you love.

NetBeans IDE is also consistently ahead of the curve in providing support for new and evolving standards, such as the new language features that were introduced in J2SE 5.0, and new specifications in all areas of Java technology.

NetBeans IDE provides an astonishing array of features right out of the box. A fully featured J2EE 1.4 development environment is built in. All the editor, debugger, and project support that is available for Java application development is also available for J2EE development. In addition, NetBeans IDE provides access to the Java BluePrints Solutions Catalog and the ability to install them as NetBeans projects, so the IDE becomes a learning tool.

The Mobility Pack enables J2ME developers to design, develop, and debug Java 2 Platform, Micro Edition (J2ME™) MIDlets from within NetBeans IDE. Providing one of the most powerful sets of mobile development tools, the Mobility Pack includes the J2ME Wireless Toolkit (WTK), a flow designer to visually layout the application logic, a screen designer to create the user interface, an integrated device fragmentation solution, and tools for building client-server applications.

NetBeans IDE features

While the NetBeans IDE modular architecture allows for plug-ins, the scope of features included provides a full-featured development environment for all types of Java applications. All the necessary modules are available and are tested together, so that NetBeans is ready to go as soon as it is installed.

The IDE's basic job is to make the edit-compile-debug cycle much smoother. For example, the IDE:

- Identifies coding errors almost immediately and marks them in the Source Editor.
- Speeds coding with code completion, word matching, abbreviation expansion, and fix import features.
- Provides visual navigation aids, such as the Navigator window and “code folding,” as well as numerous keyboard navigation shortcuts designed especially for Java programmers.

- Can display documentation for a class as you are typing in the Source Editor.
- Hot links compilation errors in the Output window, so you can jump straight to the source.
- Manages package names and references to other classes. When classes are renamed or moved around, the IDE identifies places in the code that are affected by these changes so you can have the IDE generate the appropriate changes to those files.
- Offers many debugging features that provide a comprehensive view of the way your code is working as it runs. Set breakpoints, which persist from session to session, and keep your code free of clutter.
- Helps integrate other parts of your workflow, such as checking sources into and out of a version control system. NetBeans supports Concurrent Versions System (CVS), Project Version Control System (PVCS), and Visual SourceSafe (VSS) standards, providing advanced code management features.
- NetBeans performance and memory profiler augment traditional edit-compile-debug cycle with performance profiling.
- Besides being an IDE, NetBeans is certified as a 100% Pure Java open source platform. Develop plug-in modules for NetBeans IDE, or create an entirely different application built on top of the core modules that make up the platform.

Support for Java 2 Platform Enterprise Edition

The J2EE technology defines the standard for developing multitier enterprise applications. It simplifies enterprise applications by basing them on standardized, modular components; providing a complete set of services to those components; and handling many details of application behavior automatically, without complex programming. The J2EE platform is

targeted to developers who want to write distributed transactional applications for the enterprise and leverage the speed, security, and reliability of server-side technology.

NetBeans IDE 4.1 introduces comprehensive support for the J2EE developer. Advanced wizards create J2EE components such as Web applications, servlets, JavaServer Pages™, enterprise Java Modules, Enterprise JavaBeans components, and Web services. Also, a complete runtime environment based on the Java System Application Server 8.1 is included. It is a J2EE 1.4 compliant application server that is free for development, deployment, and redistribution, and offers the ideal IDE companion for developers who need an integrated environment where complete J2EE applications can be developed, built, assembled, deployed, and debugged.

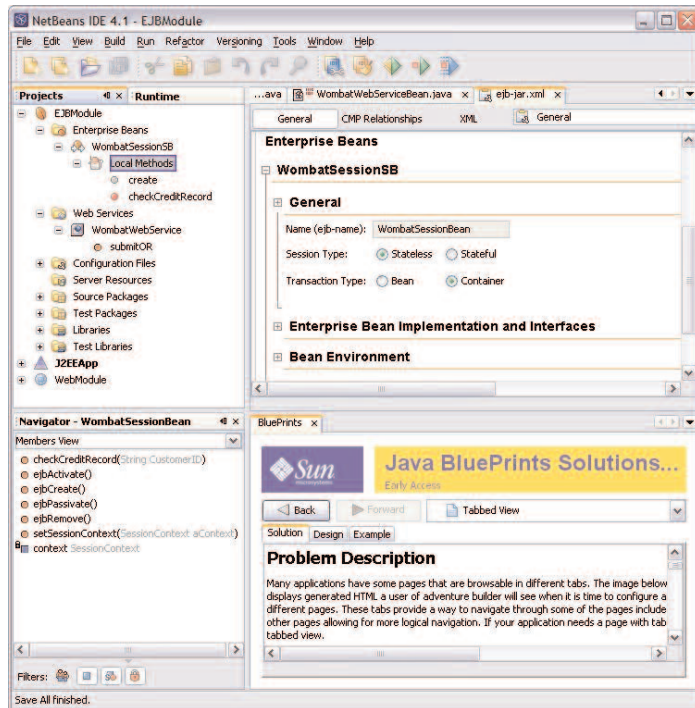


Figure 1. The upper half of the window shows an example of the user-friendly, EJB deployment descriptor editor. The lower half is an example of browsing the Java BluePrints Solutions Catalog.

The NetBeans IDE makes it simple to:

- Create a J2EE application, automatically add EJB and Web modules, and deploy the application.
- Create an EJB module and deploy it as a stand-alone module or packaged in a J2EE application.
- Create Session Beans, Entity Beans and Message-Driven Beans.
- Create Entity Beans using an existing database schema.
- Create a Web module, generate calls to EJB components, and deploy the Web module either as a stand-alone Web application or packaged in a J2EE application.
- Create, register, and test Web services.
- Create, browse, and edit database structures using the Database Explorer.
- Try out the sample applications to learn and understand J2EE technology.
- Import existing J2EE projects.
- Validate applications using the J2EE Verifier.
- Visually configure EJB components, Web Services, and Web Components.

- Add multiple source folders to an EJB or Web module, create unit tests as a part of the project.

Mobility

NetBeans Mobility Pack is a powerful tool for experienced mobile Java developers as well as those new to J2ME development, providing a quick start to development even if you have only J2SE programming experience.

The NetBeans Mobility Pack may be used for writing, testing, and debugging applications for Java technology-enabled mobile devices. This release introduces drag-and-drop visual authoring for managing application flow and screen content, end-to-end mobile application wizards, and new remote server deployment options.

NetBeans Mobility Pack also simplifies coding with templates for MIDlet and MIDlet suites, plus code completion against J2ME packages.

It solves device fragmentation problems with editable custom configurations for each device. You can easily integrate third-party emulators for a robust testing environment.

NetBeans Mobility Pack Features:

- Visual Mobile Designer — Drag-and-drop objects into the Flow Designer, Screen Designer, and Action Editor to automatically generate Java code for rapid development and prototyping.
- Wireless Connection Wizard — Easily access Web services and other server-side data from MIDlets via servlets.
- New Deployment Options — Upload MIDlets via WebDAV, File Transfer Protocol (FTP), and Security Control Protocol (SCP) from within the IDE.
- Project support built on Apache Ant organizes MIDP source code and project metadata for easier coding and management.
- Integration with the J2ME Wireless Toolkit 2.2, providing support for advanced development

- features such as MIDlet signing, certificate management, integrated over-the-air (OTA) emulation, push registry emulation, Wireless Messaging and Multimedia APIs, and Wireless Messaging API (WMA) emulation features for Short Message Service (SMS) and Cell Broadcast Service (CBS) messages.
- Device fragmentation solution — Develop applications for multiple devices by adding and executing device-specific code as configurations within a single application.
- Support for Mobile Information Device Profile (MIDP) localization — Add locale-specific components to your application.
- Support for J2ME MIDP 2.0 and Connected Limited Device Configuration (CLDC) 1.1 standards.
- Supports adding third-party emulator platforms — Test your application on a variety of target device emulators by leading manufacturers.

- Integrated obfuscation and optimization support — Adds security, improves performance, and optimizes the size of your application.
- OTA download testing — Emulator simulates behavior of real mobile devices.

Profiler

The NetBeans Profiler is a full-featured profiling tool, providing CPU, memory, and thread profiling functionality for the NetBeans IDE.

Highlights

Innovative approach to profiling:

As the size and complexity of Java applications grow, keeping their performance at the required level becomes progressively difficult. The Profiler uses a unique approach to minimize profiling overhead, which allows it to provide accurate results in situations where other profilers fail.

Tight IDE integration:

We believe that profiling should become a natural part of the development work cycle, and many more developers should invest time in understanding the memory and performance aspects of their Java applications. Other profiling tools, which are often expensive as well as difficult to set up and use, fail to address the needs of the broad developer community. To address that need, we aim to make profiling an integrated part of the development workflow, as easy to use and unobtrusive as possible.

Features

- *Low overhead profiling:* With NetBeans Profiler, you can profile just a subset of the application code — the rest of the code is not instrumented, so it runs at full speed. The profiled subset and the type of profiling (CPU, memory, and so on) can be changed at any moment during run time.

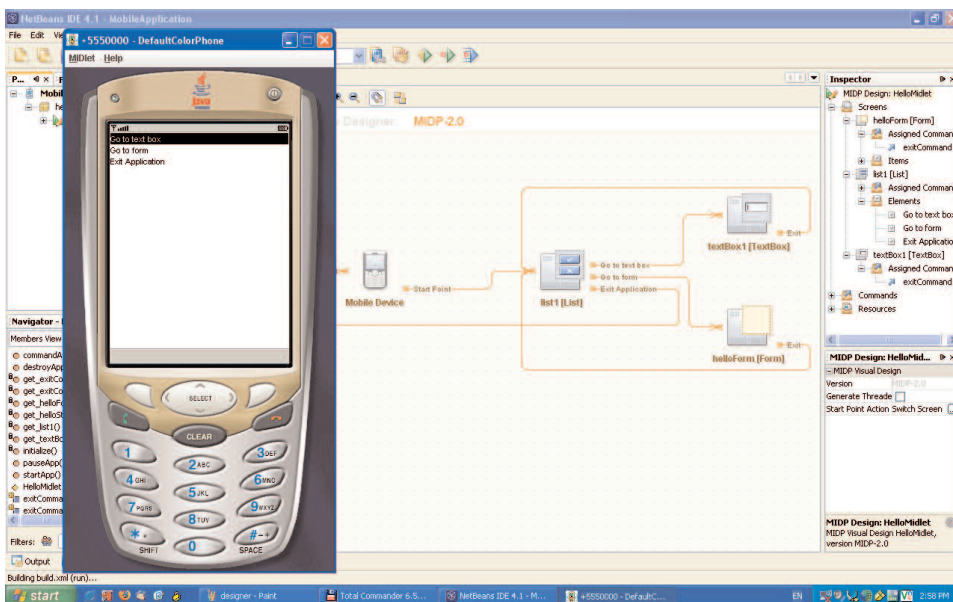


Figure 2. The top window shows one of the emulators running a J2ME application. In the background, the Visual Mobile Designer is partially visible.

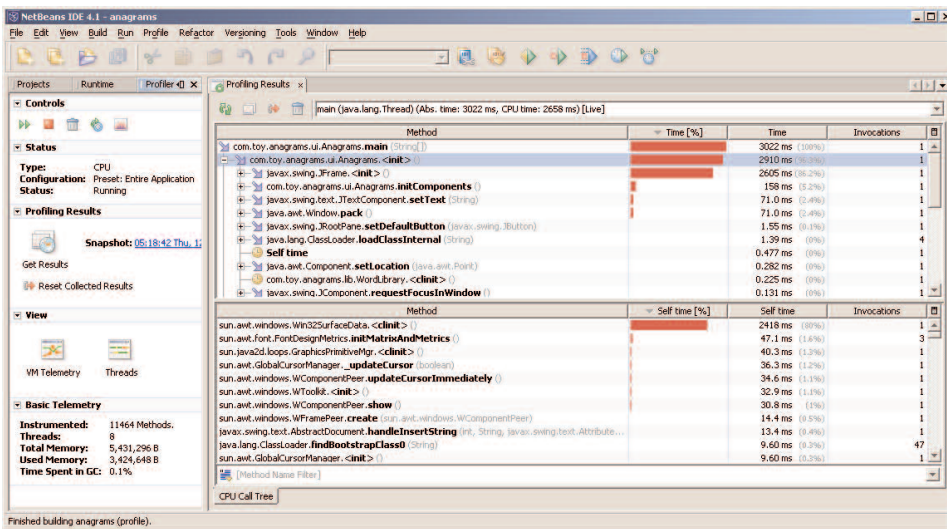


Figure 3. On the left, the Profiling Console provides shortcuts for the most common operations. On the right, two views of CPU usage are shown — the top starts from the method selected for profiling, and the bottom is a list of all methods.

- Memory leak debugging:** Memory leaks are not uncommon with large Java applications. With most existing tools, it can take hours to make sure there is really a memory leak, and identify leaking objects. Our technology can pinpoint leaking objects immediately, based on a special pattern of allocations and reclamations (or lack thereof) that such objects typically demonstrate.
- Task-based profiling:** By definition, profiling is a complex activity. Our task-based approach makes it easy by offering several of the most common tasks, such as standard CPU or memory profiling, or simple monitoring.
- Threads profiling:** Threads and their activity during application execution can be observed. Threads profiling offers two views of the threads: “Threads timeline” which shows threads activity side-by-side, and “Thread details” which shows detailed information about one or more selected threads, including a list of all state changes in the thread life.
- Attaching to running applications:** The profiled application can be started as usual, without special command line options. You can still attach and detach the profiler to/from it at any time. When the profiler is detached, there is no instrumentation in the application, and thus no overhead.
- CPU Performance profiling:** Profile the entire application, or select a subset of code based on one or more *root methods*. Selective profiling, based on chosen roots, reduces performance overhead and helps obtain accurate results for heavily multithreaded applications under realistic workloads. The results, which contain details about only the selected code, are much easier to analyze and understand.
- Memory profiling:** Only object creation events can be profiled, which imposes smaller overhead and allows you to detect problems with excessive object allocations. Additionally, liveness profiling registers both object creation and garbage collection events, allowing you to see the number, age, allocation paths, and other characteristics of objects currently in memory.

Eclipse Import

To address developer needs in migrating to NetBeans, an optional module called the Eclipse Project Importer automatically imports Eclipse projects into NetBeans as NetBeans projects, processing Eclipse project metadata and mapping it directly to a new NetBeans project.

When pointed to an Eclipse workspace, the NetBeans Eclipse Project Importer automatically detects project dependencies and suggests migrating of all dependent projects.

NetBeans™ IDE 4.1

System requirements

Operating system

NetBeans IDE runs on operating systems that support the Java Virtual Machine. The NetBeans IDE has been tested on these platforms:

- Microsoft Windows XP Professional SP2
- Microsoft Windows 2000 Professional SP3
- Solaris OS 9, 10 (SPARC® and x86 Platform Editions)
- Red Hat Enterprise Linux 3
- Java Desktop System 2
- Mac OS X 10.3

Version control systems

- CVS 1.11
- PVCS 7.5
- Microsoft VSS 6.0

Database servers

- IBM DB2 8.1
- Oracle 9*i*
- PointBase 4.2
- Microsoft SQL 2000 SP3
- MySQL 4.x

Application servers

- Sun Java Application Server Platform Edition, 8.1
- Apache Tomcat 5.5.xx
- JBoss 4.0
- WebSphere 6.0
- WebLogic 9.0

Recommended hardware configuration

- Solaris™ operating system:
 - Processor: 500 MHz Ultra™ 60 workstation or equivalent
 - Memory: 512 megabytes
 - Disk space: 125 megabytes of free disk space
- Microsoft Windows operating systems:
 - Processor: 780 MHz Intel Pentium III workstation or equivalent
 - Memory: 512 megabytes
 - Disk space: 125 megabytes of free disk space
- Linux operating system:
 - Processor: 800 MHz Intel Pentium III workstation or equivalent
 - Memory: 512 megabytes
 - Disk space: 125 megabytes of free disk space

Learn More

Get the inside story on the trends and technologies shaping the future of computing by signing up for the Sun Inner Circle program. You'll receive a monthly newsletter packed with information, plus access to a wealth of resources. Register today at sun.com/joinic.