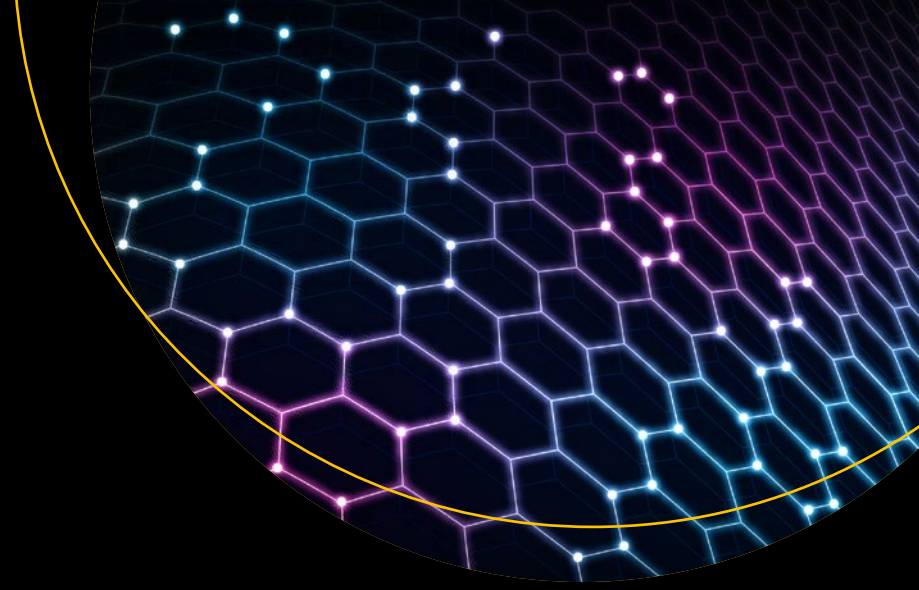


Pro Apache NetBeans

Building Applications on the Rich Client Platform

—

Ioannis Kostaras
Constantin Drabo
Josh Juneau
Sven Reimers
Mario Schröder
Geertjan Wielenga

Apress®

Pro Apache NetBeans

Building Applications on the Rich Client Platform

Ioannis Kostaras
Constantin Drabo
Josh Juneau
Sven Reimers
Mario Schröder
Geertjan Wielenga

Apress®

Pro Apache NetBeans: Building Applications on the Rich Client Platform

Ioannis Kostaras
The Hague, South Holland, The Netherlands

Constantin Drabo
Ouagadougou, Burkina Faso

Josh Juneau
Chicago, IL, USA

Sven Reimers
Salem, Germany

Mario Schröder
Berlin, Germany

Geertjan Wielenga
Amsterdam, The Netherlands

ISBN-13 (pbk): 978-1-4842-5369-4
ISBN-13 (electronic): 978-1-4842-5370-0
https://doi.org/10.1007/978-1-4842-5370-0

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Jonathan Gennick
Development Editor: Laura Berendson
Coordinating Editor: Jill Balzano

Cover image designed by Freepik (www.freepik.com)

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail rights@apress.com, or visit http://www.apress.com/rights-permissions.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at http://www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/9781484253694. For more detailed information, please visit http://www.apress.com/source-code.

Printed on acid-free paper

Dedicated to our families who have silently supported us in our adventures as authors. More specifically:

Ioannis Kostaras:

To my lovely wife Katerina; my adorable little son Nikolaos-Ioannis; and my parents, Nikolaos and Zinovia

Constantin Drabo:

I dedicate this book to my lovely mother, my brother Emmanuel, and sister Denise, my spouse Mariam, and my little son Johann. Finally, to these special persons: Geertjan Wielenga and Malo Sadouanouan

Josh Juneau:

To my wife Angela; and my children, Katie, Jake, Matt, Zach, and Luke

Sven Reimers:

To my beloved wife Isabelle

Mario Schröder:

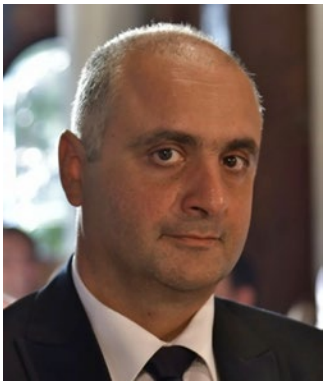
To my wife Fridah

Geertjan Wielenga:

To my wife Hermine

Table of Contents

About the Authors



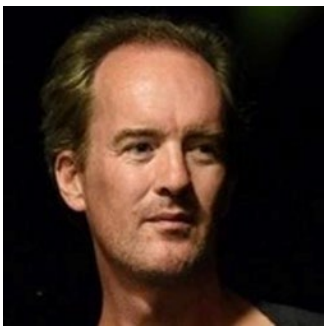
Ioannis (John) Kostaras graduated from the Informatics Department at the University of Athens. He also holds a Master of Science degree in Telecommunications and Information Systems from the Electronics Systems Engineering Department of Essex University, and a Master of Arts degree in Management Studies also from Essex University. Ioannis is a software architect/senior software engineer in an international organization located in The Hague, Netherlands, developing real-time critical distributed systems, and has been a Java developer since JDK 1.0 was released. One of the many applications he has written for the industry, developed in the NetBeans Rich Client Platform, was awarded the 2012 Duke's Choice Award. He is also a member of the NetBeans Dream Team and an Apache NetBeans contributor. Apart from Java, he "speaks" other languages, such as C/C++, Python, Ruby, and Erlang, to name a few; and he codes them using his favorite IDE, NetBeans. He has written a number of articles in JavaCodeGeeks, and in the NetBeans wiki, and runs an online course for Java in Greek in the Mathesis platform, teaching the latest Java and Apache NetBeans to audiences speaking Greek. Finally, but not least, he is a co-organizer of JCrete, the hottest Java unconference, on the island of Crete.

Constantin Drabo earned a Master's Degree from the Université de Picardie Jules Verne (France) in Internet Technologies. He is a software engineer in the National Treasury of Burkina Faso, and is a part-time teacher at the Université Nazi Boni of Bobo-Dioulasso, where he teaches on Java/Java EE and the Human/Machine Interface. He participates in the Java Community Process (JCP) through his contribution to JSR 381 (Visual Recognition). Constantin is a member of the NetBeans Dream Team, and creator of the FasoJUG Java User Group. He is currently studying for his PhD in Machine Learning in Medical Systems.

Josh Juneau has been developing software and database systems for several years. Database application development and sophisticated web apps have been the focus of his career since the beginning. Early in his career, he became an Oracle database administrator and adopted the PL/SQL language for performing administrative tasks and developing applications for the Oracle database. In an effort to build more complex solutions, he began to incorporate Java into his PL/SQL applications, and later developed standalone and web applications with Java. Josh wrote his early Java web applications utilizing JDBC to work with back-end databases. Later, he incorporated frameworks into his enterprise solutions, including Java EE, Spring, and JBoss Seam. Today, he primarily develops enterprise web solutions utilizing Java EE. He is an avid contributor to Oracle's *Java Magazine*, speaks at conferences and Java user group events, contributes to the Chicago Java User Group, and is an Apache NetBeans committer and a Java Champion.

Sven Reimers is based at Lake Constance in Southern Germany where he works as a systems engineer for Airbus Defence and Space creating next-generation ground segment software for space systems. He has more than 20 years' experience building complex distributed software systems, and more than 20 years' experience with Java. In 2009, Reimers was the winner of the Duke's Choice Award in the Network Solutions category for ND SatCom's Satellite Communication Management Software. He is part of the Apache NetBeans PMC, an author at OpenJFX, a regular speaker at conferences, as well as leader and founder of JUG Bodensee. For his long-term commitment to Java, the Java community, and his contributions to the Java ecosystem, Reimers was named a Java Champion in 2014.

Mario Schröder earned a Master's Degree in Civil Engineering from the University of Rostock in Germany. However, he discovered his passion for computer science during his studies and made programming and software development into his profession. He started working with Java professionally in 2001. He is a certified Java Programmer, iSAQB-Software Architect, and ISTQB-Tester. His first contact with NetBeans was in 2004. He was a member of the NetCat team for Version 7.2, has contributed patches to the platform, and has developed several plugins.



Geertjan Wielenga is the Apache NetBeans PMC Chair and a product manager at Oracle for Oracle JET, which is Oracle's free and open source front-end JavaScript technology stack. He lives and works in Amsterdam. He is a Java and JavaScript technology enthusiast, advocate, trainer, speaker, and writer. He studied law at the University of Natal in Pietermaritzburg, South Africa. In early 1996, he started working as a technical writer for a software technology organization in the Netherlands and, after having worked at various other software organizations in the Netherlands, he moved to Vienna, Austria, and worked as a technical writer for the documentation of Coca-Cola's ERP system. In 2004, he moved to Prague, Czech Republic, to work on NetBeans IDE, again as a technical writer. He discovered that NetBeans IDE is a unique product, project, and ecosystem and became inspired and continually enthused by the open source ecosystem around NetBeans IDE, the technologies around it, and the central role that the NetBeans community plays in them. He is happy and proud of the role NetBeans IDE has played in its many years of existence and its continued development in the Apache Software Foundation.

About the Technical Reviewer

Kevin Farnham has developed software for high-volume data analysis and mathematical modeling since the 1980s. He has been site editor and lead blogger for technology websites at Oracle, Intel, and O'Reilly Media.

Kevin has served as technical editor and reviewer for dozens of software engineering books.

Acknowledgments

Authoring a book is like departing for an adventure. Even though you make a plan, you don't know how long it is going to take, how big the waves will be, and what new land you are going to explore. Like Odysseus, we seek our Ithaca and the way is not always easy – it is full of adventures and exciting indeed – but rewarding in the end.

This book is our adventure into an IDE we all love and continue to work with. So before anybody else, we would like to acknowledge and thank this little team from the University of Prague who had the fancy idea of creating NetBeans, its never-tired Product Manager, Geertjan Wielenga, as well as all the people that have contributed one way or another, all these years, to make this IDE a necessity for all of us.

We would also like to thank the Apress team for their patience and support in a multi-author community book that has not always been easy to manage and coordinate, their persistent comments leaving nothing to chance, and scrutinizing every sentence of the drafts. However, any errors that may still exist are the responsibility of the authors.

Finally, but not least, we would like to thank our families for bearing with us during our adventure, when we were spending many hours behind a computer's screen, trying to explain as best as we could the many NetBeans topics you will find in this book, as well as writing example programs that would give more value to its readers.

Introduction

Pro Apache NetBeans provides a detailed overview of the latest additions to NetBeans IDE, like support for JShell, the Jigsaw Module System, and Local Variable Type Inference, focusing on what this new version brings to developers. It also describes new features in NetBeans Platform, that is, the framework the NetBeans IDE has been built upon.

The book is a practical, hands-on guide providing a number of step-by-step recipes that help you take advantage of the power in the latest Java (and other) software platforms, and it gives a good grounding in using NetBeans IDE for your projects. This book has been written by the Apache community members who have been using the IDE and actively contributing and developing Apache NetBeans as an open source project.

Pro Apache NetBeans consists of three parts. The first part describes how to use the IDE as well as the new features that it brings to support the latest Java versions. The second part describes how you can extend NetBeans by creating plugins or writing your own standalone applications using the Rich Client Platform. The third part describes how you can contribute to develop your favorite IDE further, becoming part of the open source team that is driving future developments in the toolset.

In more detail, Chapter 1, "What Is NetBeans," provides a gentle introduction to NetBeans. It explains what NetBeans is and what it is not, its history, all the way to its new home, the Apache Software Foundation, being its second biggest project. It also provides an introduction of the many programming languages and tools that NetBeans supports, and it introduces you to many other things that are explained in more detail in the rest of the book.

Chapter 2, "Getting Started with NetBeans," provides an overview of the NetBeans IDE, which will help new users to become familiar with its User Interface, the various windows, and it introduces you to how to build your first Java project. It also provides useful tricks and tips to help you speed up your productivity.

Chapter 3 provides an overview of the new features that Apache NetBeans IDE, since version 9.0, provides for Java versions 9–12 and other programming languages. It describes support for JShell, the Jigsaw Module System, HttpClient and Local Variable Type Inference, as well as improvements for PHP.

Chapter 4 describes how to develop desktop applications. These are standalone applications running, typically, locally on your machine without the need of a web or an application server. The chapter explains there are three ways to develop Java GUI desktop applications: using the Abstract Window Toolkit (AWT), Swing, or JavaFX. NetBeans provides a very powerful visual editor that you can use to build your GUI with dragging and dropping components onto forms (panels). The chapter also guides you through the steps for developing your own desktop application.

Chapter 5 describes how to build Enterprise (or Web) applications. Contrary to desktop applications, these require a web or application server to serve HTML pages. Java EE support has been included with Apache NetBeans since release 11.0. Full support for Maven web applications and Java EE technologies such as Enterprise JavaBeans (EJB), Contexts and Dependency Injection (CDI), and JavaServer Faces (JSF) allows developers the convenience of auto-completion, code fragments, and easy syntax recognition. After finishing this chapter, you will have a basic understanding of the conveniences provided by Apache NetBeans for full stack Java EE development.

Chapter 6, "Debugging and Profiling Applications," explains in detail the powerful debugger and profiler of Apache NetBeans. If debugging makes your code right, profiling will make it run fast. Things like breakpoints, debugging multi-threaded code, CPU and memory profiling, snapshots, etc., are explained in this chapter.

Part II of the book describes the NetBeans Platform (or Rich Client Platform), that is, the framework or libraries that the NetBeans IDE has been built upon. With this knowledge you can better understand the NetBeans source code that will allow you to fix bugs, or extend it by writing your own plugins. Or you can use it to develop your own desktop applications faster than using Swing.

Chapter 7, "Mastering the Core Platform," introduces the reader to the core APIs of the NetBeans Platform, namely, the NetBeans module system, the FileSystem, and the Lookups. NetBeans provides its own module system to modularize applications, which supports the use of OSGi. A comparison with Java's module system, Jigsaw, is provided in this chapter. NetBeans also provides its own libraries to access the filesystem. A comparison with NIO.2, introduced in Java 7, is provided. Finally, the Lookups are a very powerful tool to build loosely coupled applications. A comparison with Java 6 `ServiceLoader` is also provided.

Chapter 8, "Mastering the User Interface," teaches the reader the GUI components of the NetBeans Platform, namely the Window system, the Action system, the Node System, and the Explorer Views and Property Sheets, all reducing the time you need to develop powerful UIs.

In Chapter 9, the reader applies what s/he has learned so far by porting a desktop application to NetBeans RCP. The development is done in three steps: (1) Build a "static" visual prototype of the user interface; (2) Build a "dynamic" prototype of the application, coding user interface events and associated business logic and creating customized UI components as needed; and (3) Code the persistence logic.

Chapter 10, "Learning the Extras of the Platform," describes many of the other APIs provided by the NetBeans Platform, such as the Dialogs API, the Visual Library and the Palette, the Status Bar and Notifications, the progress bar, the QuickSearch and the Output window, etc.

Chapter 11 teaches you how to extend NetBeans by writing a plugin for it. You will learn step by step how to develop a plugin for Hyperledger, an open source effort to advance blockchain technologies, hosted by the Linux Foundation.

Part III gives you an overview of the new home that hosts NetBeans, the Apache Foundation; its infrastructure to support open source projects like the various tools that allow issue tracking, testing, documentation, etc.; and gives you all the information you need to become a contributor of Apache NetBeans.

Chapter 12 describes the Apache NetBeans Process. After a short history, it explains how the migration of source code was achieved from Oracle to Apache, the various NetBeans versions under Apache as an incubator project, and the graduation to a top-level Apache Project.

Chapter 13 describes the Apache Infrastructure and the various tools that the Apache Software Foundation provides for its projects for issue/bug reporting, wikis, websites, etc.

Chapter 14 describes the NetCAT program. The ***Net***Beans IDE ***C***ommunity ***A***cceptance ***T***esting program is very important for the quality of the product. A typical NetCAT program takes around two months of activities. The chapter explains how each NetBeans version is tested by the community before it is released. You can easily participate in this program and execute tests, thus helping to improve your favorite IDE.

Chapter 15, provides an overview of the NetBeans source code, its architecture, its various modules, and their dependencies. It describes how to download and build the NetBeans source code, load it in NetBeans IDE, and debug it. Finally, this chapter encourages you to contribute to the NetBeans source code by fixing bugs and committing your changes to a baseline.

PART I

Using Apache NetBeans

CHAPTER 1

What Is Apache NetBeans

It isn't often that a technology sweeps the world and ends up being a relevant piece of the technological universe for over 20 years. However, the Java language is one of the few that has. The Java programming language was originally developed in 1995 by Sun Microsystems, and it has grown into a massive ecosystem over the years. The Java Virtual Machine (JVM) is a virtual computer that is defined by a specification, and the Java programming language can be used to program applications that run on the JVM. A JVM can be installed on just about any computer or hardware device that is available on the market. As such, the JVM is installed on millions of devices worldwide, and there have been dozens of different programming languages built on top of the JVM that compile down to Java byte code, allowing developers to utilize different language syntax to run on the JVM. What does this mean? Well, frankly it means that the JVM and Java are everywhere, and one can use a multitude of languages to program applications that will run just about anywhere. Java is an ecosystem that started in 1995, and it will be around long into the future.

As developers on one of the most prominent development platforms, folks can develop applications in a number of different ways. It is possible to develop Java code in a text editor, compile and run it via a terminal using the `java` executable, and then be on your way. It is also possible to create libraries of files and projects all via the terminal, without the use of any specialized tools for development. However, although it can be done, development without a code editor or a "development environment" including a project-based system, can be cumbersome, and Apache NetBeans is one of the most popular development environments for the Java Platform that helps to make development easy. Although Apache NetBeans is a great development tool, it is much more than just an editor. Apache NetBeans is a comprehensive development tool with support for several languages; it is a development platform for creating rich client applications; and it is a vibrant open source community with members from around the globe.

I. Kostaras et al., *Pro Apache NetBeans*, https://doi.org/10.1007/978-1-4842-5370-0_1

The Journey to Apache NetBeans

In 1996, a student project was started under the guidance of the Faculty of Mathematics and Physics at Charles University in Prague. This student project aimed to make the development of Java much easier, providing a syntax editor for the code that would help to reduce errors and speed up development time when programming. It allowed one to compile and build entire projects with just a few clicks of a button. This project was originally named Xelfi, and in 1997 a company was formed around the project and it was renamed to NetBeans IDE. Sun Microsystems liked the project so much, that they purchased it in 1999 and open sourced the NetBeans IDE a year later.

After Sun Microsystems purchased Apache NetBeans, the user base and community began to thrive. Primarily developed and maintained by Sun Microsystems at the time, the community was able to provide patches, enhancements, and so forth since the IDE was open source. However, the overall road map and direction of NetBeans was controlled by the commercial entity. That wasn't necessarily a bad thing, as it helped to foster the IDE and the Java ecosystem as a whole, since NetBeans was being targeted toward both beginners and experienced Java developers alike.

NetBeans was part of the portfolio that Oracle had purchased from Sun Microsystems in January of 2010. As part of that portfolio, Oracle strived to drive NetBeans forward as the first IDE that would support the latest features of Java SE and Java EE alike. Oracle also strove to make NetBeans an IDE for many languages; they even hired Python and Ruby developers to help foster the JVM counterparts of those languages, Jython and JRuby respectively. As part of that initiative, NetBeans IDE gained support for these languages via the use of separately installed NetBeans modules. Since NetBeans was developed as a modular IDE, it was easy to add functionality as time went on, without increasing the footprint of the core IDE.

Oracle did a great job of moving NetBeans forward: it was still an open source IDE, and the community also played an important role. However, Oracle still drove the road map, and it was not very straight forward to become involved contributing code to NetBeans under the Oracle open source process. The NetBeans community has developed conferences organized around the IDE. Many have developed modules to extend the IDE. Even more have given talks or written tutorials about NetBeans features. The next step in the evolution of the IDE was to make contribution easier, and to give the IDE road map and sources to the community.

Oracle donated the NetBeans IDE sources and tutorials to Apache in the fall of 2016. Even though Oracle donated the IDE to Apache, it still remains very much involved in the development of Apache NetBeans, but the community now controls the road map. Contributing to Apache NetBeans is now as easy as creating a GitHub Pull Request. Since the donation, Apache NetBeans has changed its release cadence so that its releases are now more closely targeting the Java release cycle. Apache NetBeans still continues to be the first IDE to support the latest flavors of the JDK, and the community is as vibrant as ever.

Installing Apache NetBeans

There are a couple of different ways to install Apache NetBeans onto your machine, depending upon your uses. For those who are interested in modifying Apache NetBeans, the IDE can be built from source after cloning the Git repository. For those who are simply interested in utilizing Apache NetBeans as a development tool, the archive can be downloaded and extracted, and then the executable can be invoked. This section will briefly explain how to install by extracting Apache NetBeans. If you are interested in building from source, please refer to Chapter 14.

The initial requirement for utilization of Apache NetBeans is the installation of a Java Development Kit. Apache NetBeans supports JDK 8 and 11. The JDK must be installed in order to build the IDE from source or to run the IDE. To download Apache NetBeans, visit the downloads area at `https://netbeans.apache.org/download/` and obtain the most recent release. The downloads page includes links to obtain the sources or binaries for each of the releases. It also contains information about building the IDE from source. To download the binaries, click on the "Binaries" link, and then the "Apache Download Mirrors" page will be displayed. Choose a download mirror and click to obtain the archive.

Once the archive has been downloaded, extract within a folder. The extracted archive should resemble that in Figure 1-1. The executable is contained within the bin directory.

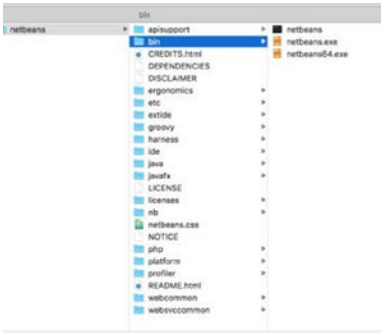


Figure 1-1. *Apache NetBeans Installation*

If more than one Java installation exists, edit the `etc/netbeans.conf` file accordingly and add the `JAVA_HOME` that should be used for running Apache NetBeans. To run Apache NetBeans, double-click the executable for the corresponding workstation's platform, which resides within the `bin` directory.

Note To learn how to build Apache NetBeans from source, please refer to Chapter 14.

Integrated Development Environment

Apache NetBeans is most widely known as an Integrated Development Environment (IDE). That is why it is commonly referred to as NetBeans IDE. Originally made to be a development environment for Java, the breadth of Apache NetBeans IDE has grown immensely over the years into a development tool for many different languages. As of the release of Apache NetBeans 10, the IDE contains first-class support for a variety of languages, including Java, PHP, and JavaScript. Support not only means that the editor contains nice syntax coloring or keyword highlighting, but it also typically contains features such as automatic indentation as per language standards, auto-completion for keywords and code fragments, and in some cases also automated rebuilds of projects for an easy debugging and development life cycle. The source code editor makes it easy to jump to different portions of code using keyboard shortcuts and mouse clicks. The editor font, color, etc., can be modified within the Apache NetBeans preferences.

Did I mention debugging? Yes, the development environment also contains a full debugger for various languages, including the ability to set breakpoints, review variable values during live debugging, and perform hot code refreshes. The development environment also contains a full-fledged performance tuning solution for Java applications. One can use the performance tools to run an application and pinpoint code issues or threads that may be causing problems.

When it comes time to build a project, few build frameworks that are as widely used as Apache Maven. The Apache NetBeans IDE contains outstanding support for Apache Maven, making a sometimes daunting build framework easy to use. If one chooses to utilize XML directly for pulling in Maven dependencies, then the XML editor is there to assist through the process, but if one would rather simply type in required dependencies and have them automatically added, then that is another option.

One of the most important pieces of the development process is workflow and team collaboration. The NetBeans IDE contains first-class support for version control systems such as Git and Subversion, making collaboration a breeze. It is possible to clone Git repositories, update code, commit changes, and push to remote repositories right within the IDE. It is also easy to compare differences between files using the DIFF comparison tool.

Over the next few sections, we will go through some of the basic Apache NetBeans IDE features for some of the most common use cases.

Java Code Editor

First and foremost, Apache NetBeans IDE is a world-class Java editor. The editor contains a number of features to assist developers, making all aspects of application development more productive. Although this section highlights Java code editing features specifically, it should be noted that many of these same features are available for other languages, such as JavaScript and PHP.

Code completion provides developers with a boost by making code easier to write with less keystrokes, and also because it is great for typing those method names and variables that you don't quite remember. The code completion is automatically invoked by pressing `CTRL-Space` while typing. Doing so will cause a drop-down menu to appear, which includes a number of code suggestions for continuing your current line of code. The code suggestions are context-based, providing meaningful suggestions for the current code at the top of the list, and less meaningful suggestions toward the bottom.

Of course, code completion is configurable, so it can be made to automatically open the code completion drop-down menu while typing or when typing certain characters. By default, when a "." character is typed, context-based suggestions will appear, which can be beneficial for easily completing method names and so forth.

Code templates and snippets are handy for inserting segments of skeleton code, as needed, making it easier to complete logic. Templates can be invoked by clicking on a suggestion icon or by typing various strings of code such as "sout". When "sout" is typed and then the spacebar is pressed, the code "System.out.println("")" is expanded, leaving the cursor within the parentheses to continue typing. One can define custom code snippets for code that is often used. There are also refactoring options for automatically completing simple boilerplate code, such as creating getters and setters, without coding at all.

Java SE Applications

Apache NetBeans was originally built for development of Java Standard Edition (Java SE) applications: that is, a Java application that contains one or more Java classes, a starting point, and an action or graphical user interface that provides functionality for a user. Traditional Java development requires Java classes to be compiled before they can be executed. Compiling and execution of an application typically occurs via a command prompt or terminal, utilizing the `javac` compiler and the `java` command to execute resulting class files. What's more, if an application consists of more than one class file or resource, then it must be packaged into a *Java Archive (JAR)* before it is executed. Performing these tasks manually can be time consuming and cumbersome. Why not automate these minuscule tasks so that the developer can spend more time creating, rather than performing routine tasks? That is the original intention of Apache NetBeans IDE … to make development easier and more productive by reducing the number of required routine tasks.

Note Prior to installing Apache NetBeans, be sure to have a JDK installed. If you are planning to develop Java EE/Jakarta EE applications, be sure to install the version of the JDK that you will be using for the container to which the application is deployed. Once installed, the netbeans_jdkhome configuration variable, located within etc/netbeans.conf, should be assigned the path to the JDK you are using to run Apache NetBeans.

Let's walk through the development of a simple "Hello World" Java application and introduce some of the core features of Apache NetBeans IDE. To begin, open Apache NetBeans and create a new project by choosing "File" ➤ "New Project" or selecting the "New Project" icon in the toolbar.

Next, within the "New Project" dialog, choose the "Java" category from the *Categories* pick list, and then select "Java Application" from the *Projects* pick list. A *New Java Application* dialog will appear (Figure 1-2), and a **Project Name**, **Project Location**, and **Project Folder** can be provided. Type "HelloApacheNetBeans" as the project name.

Next, select a location on your computer in which to store the project files. Apache NetBeans allows one to provide a common area for storing third-party libraries, but skip that for now, and ensure "Create Main Class" is checked, maintaining the default *Main Class* name. Click "Finish" to create the project.

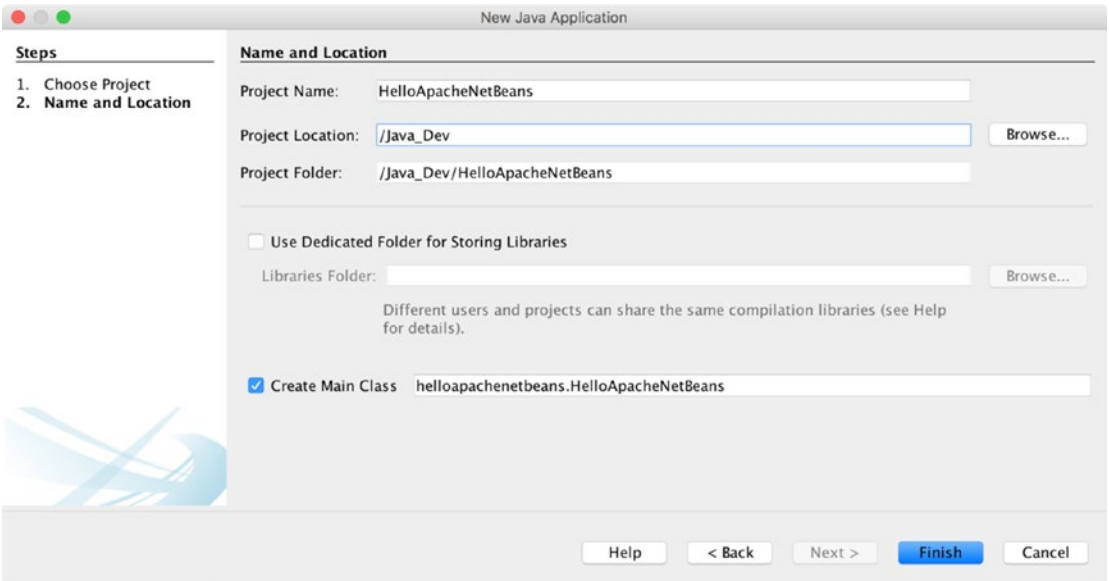


Figure 1-2. *New Java Application Dialog*

Apache NetBeans will open the `HelloApacheNetBeans` main class in the editor, and you will see that the project has been added to the left-hand navigator (Figure 1-3). When the project menu in the left-hand navigator is expanded, a single source file named HelloApacheNetBeans.java is displayed within the package `helloapachenetbeans`, and the default JDK that is registered with Apache NetBeans can be seen under the Libraries node. The HelloApacheNetBeans.java file is opened in the source editor by default.

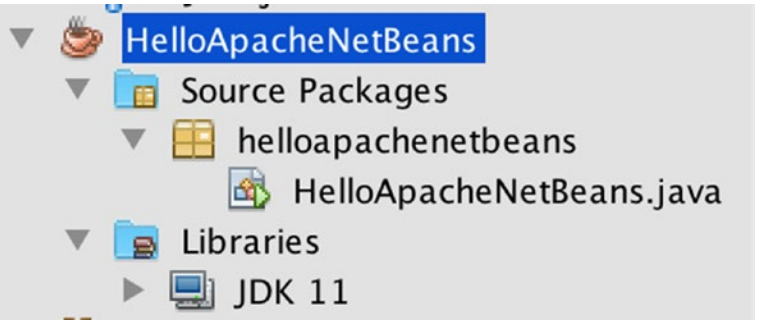


Figure 1-3. *Project Menu*

The project is displayed in an enhanced format, making it easy to navigate the project files and see project dependencies. At the top of the left-hand navigator, there are tabs for "Project," "Files," and "Services." The "Project" tab is displayed by default, and it will contain each of the currently open Apache NetBeans projects in the enhanced view. The "Files" tab shows each of the currently open Apache NetBeans projects using the actual project files as they can be seen on disk. Sometimes this can be useful if you know where a certain file is placed and you need to edit it. It also can be useful because Apache NetBeans will hide some of the project files by default to organize the project and make navigation easier, and also because these hidden files should not need to be edited in most cases. The "Services" tab contains a multitude of options for working with databases, web services, Maven repositories, Docker, etc. Many of these various services will be covered later in this book in detail.

To get an opportunity to work within the editor, place the cursor within the `main()` method of the `HelloApacheNetBeans.java` file in the open editor, after the code comment that reads as follows:

```
// TODO code application logic here
```

Press the return button to start a new blank line, and begin typing: `System.out.println("Hello Apache NetBeans");` While typing, auto-completion options are presented to make coding easier. In fact, by simply typing `sout` and pressing the tab button, the line will be auto-completed using a code template. Type the string "Hello Apache NetBeans" within the `println()` method. The code should look like that below:

```
public static void main(String[] args) {
  // TODO code application logic here
  System.out.println("Hello Apache NetBeans");
}
```

Within the project navigator, the right-click provides many additional options, as it brings up the contextual menu. Most of these options will be covered later in the book, but for now right-click on the HelloApacheNetBeans project within the project navigator and select "Clean and Build." Doing so removes any compiled code from disk, and performs a build including any external dependencies for the project. Next, right-click on the project name and click "Run." Below the editor resides a number of tabs, and one of them is the "Output" tab. When the project is run, the "Output" tab should display the following:

```
run:
Hello Apache NetBeans
BUILD SUCCESSFUL (total time: 0 seconds)
```

Java Web Applications

Java web applications typically have a number of processes that need to be coordinated in order to develop and test. For instance, most web applications need to be deployed to an application server container of some kind, and many access a database. Apache NetBeans allows for easy development, deployment, and testing of Java web applications by allowing one to maintain all of these separate processes within one place. What's more, it contains a debugger mode that will easily allow one to deploy an application in debug mode and then step through breakpoints to help troubleshoot issues. See Chapter 5 for more details on the debugger.

Java web applications typically contain a number of different code and styling files, and the Apache NetBeans code editor is capable of editing just about any file. The code editor supports many different languages with features such as syntax highlighting, auto-completion, and error marking. When working with web applications especially, having the ability to edit different types of files within the same editor is essential. The Apache NetBeans plugin portal also provides the ability for developers to create third-party extensions for the IDE, making the possible supported files and options for expansion unlimited. For instance, there are third-party plugins available for many application server containers, allowing special functionality to be made available for these servers. See Chapter 11: Writing a Plugin for NetBeans for more details.