

CHAPTER

3

Setting Up Your Environment

In this Chapter

This chapter describes how you choose, download, install, and configure the software that you need to write and compile Java source files. This topic—generally referred to as *setting up your environment*—is included in an appendix in most books on the Java programming language. We have found in our trainings, however, that failing to set up your environment properly can be a major roadblock in learning how to write developer documentation; thus, we cover setting up your environment in a chapter.

Unless your environment is already configured correctly, we suggest you go through the material in this chapter carefully to ensure a trouble-free installation and configuration.

In addition, this chapter describes how to download and install the sample Java code projects used in the book.

What Software Do You Need?

To write practice API documentation for the projects in this book, you will use:

- A text editor

You need an authoring tool to open the Java source files and add Javadoc documentation comments. See “Choosing a Text Editor” later in this chapter, which discusses the pros and cons of various authoring software.

- The Java 2 Platform Standard Edition (J2SE) Development Kit Version 5.0

You need the Java programming language, which is contained in the Java 2 Platform Standard Edition (J2SE) Development Kit Version 5.0 and available as a free download from the Sun Microsystems web site.

Note While the current version of the Java 2 Platform Standard Edition (J2SE) Development Kit is Version 5.0 Update 7, Java SE 6 is in Beta. To check for the latest instructions for downloading the JDK or other software referred to in this chapter, please go to our web site at <http://www.bitzone.com> where we maintain the latest version of this chapter.

- The Ant build tool

While you can run the Java compiler by itself, we have included a simple Ant *build script* with the sample code projects to introduce you to a build tool. All real-world software projects use a build tool.

Using a build tool is analogous to using macros in, say, a word processing application to automate repetitive editing and formatting tasks. Using a build tool ensures that any mutual dependencies between *modules* happen consistently and in the right order. You may not know the order in which the modules need to be

compiled by merely looking at the file directories; you can automate the order in which the modules compile using Ant. (According to the Apache web site, *Ant* stands for *Another Neat Tool*.)

- Command Prompt window

You will be running Ant from a command line interface (CLI), so you also need a Command Prompt window readily available. You may already have a desktop shortcut to a Command Prompt window. If not, you can open one by choosing Accessories > Command Prompt from the Windows Start menu.

A JDK by Any Name...

In the decade since Sun Microsystems first released it, the Java platform has gone through dozens of name changes. What has remained constant is that the developers who actually use Java to do their work refer to the two major components of the Java platform by the acronyms *JRE* (Java Runtime Environment) and *JDK* (Java Development Kit). The *JRE* allows the computer on which it is installed to run programs developed in Java, and the *JDK* allows programmers to develop those programs. In the book from this point on, we will use the term the *JDK* or the *JDK 5.0 Update 7* to refer to the Java Standard Edition Development Kit Version 5.0 Update 7 unless referring to a specific link.

When referring to a specific link or other identifying feature on the Sun Microsystems web site, we use the specific term as seen on the web site. As this book goes to press, Sun Microsystems is making a transition from referring to the Java Platform, Standard Edition as *J2SE* to referring to it as *Java SE* on the Sun Microsystems web site. The terms *J2SE* and *Java SE* both refer to the Java Platform, Standard Edition.

Documenting APIs: Writing Developer Documentation for Java APIs and SDKs

The JDK

The JDK includes a number of components and is available on a number of platforms, as described in this section.

COMPONENTS IN THE JDK

The JDK includes the following components:

- The Java Runtime Environment (JRE), which includes:
 - The Java Virtual Machine (JVM)
 - Class libraries
- The JDK development tools, which include:
 - javac (the compiler)
 - javadoc (the Javadoc API documentation tool that generates a reference guide)
 - Other tools. (See Appendix C, “Java Tools,” for more information about the JDK tools.)

SYSTEM REQUIREMENTS FOR THE JDK

The JDK is available for the following platforms:

- Windows
- Macintosh
- Linux
- Solaris

For Windows, the JDK runs on the following operating systems:

- XP Professional
- XP Home

- 2000 Professional
- 98
- ME

The amount of disk space required for a complete installation of all components of the JDK on a PC is approximately 264 MB.

This chapter provides instructions for the Windows platform only. The Macintosh OS X comes with a certified release of the Java Platform, Standard Edition. For more information, consult your Macintosh OS X Help or visit the Apple Developer Java web site at <http://developer.apple.com/java/>.

The Ant Build Tool

As mentioned earlier, *build tools* are used in software projects to automate repetitive and complicated tasks, such as compiling multiple source files stored in different directories. Ant is a popular build tool that was developed as part of the Apache Jakarta Project. To download the latest version of Ant (available without charge) go to <http://ant.apache.org/>.

Other commonly used build tools are **make**, the Unix build tool, and **nmake**, the Windows/DOS version of this tool. The makefiles (the equivalent of Ant build scripts) for both of these tools are difficult to write because the syntax for the makefiles is arcane. For example, a space and a tab have different semantics.

In addition to having complex semantics, these build tools are shell-based and therefore not platform independent. In contrast, because Ant is written in the Java programming language, it runs under any operating system that supports Java.

Another plus for Ant is that its build files are written in *XML* and are fairly easy to interpret and create; in contrast, even veteran programmers try not to write build files for **make** but rather modify existing build files for their particular needs. Thus, the

authors have selected Ant for the projects in this book because of its popularity, platform independence, and ease of reuse for the build files.

Choosing a Text Editor

One of the most helpful tools for writing API documentation is a *full-featured text editor*. Such editors include, for example, *syntax color highlighting* that displays the constituent parts of the code in different colors and fonts. This feature facilitates identifying errors in Java source code.

We suggest you *not* choose FrameMaker, Word, or Notepad as your preferred text editor, even though, as a technical writer, you may already own them and be proficient in their use. In addition, even if you have experience with an IDE (Integrated Development Environment), we suggest you *not* use it either.

IDEs use wizards and drag-and-drop functionality that insulates a developer from the source code and conditions a developer into certain habits. Using only a text editor and a build tool such as Ant, on the other hand, provides the thinnest layer between you, the student, and the code and thus maximizes your learning experience.

WHY NOT USE FRAMEMAKER? OR EVEN WORD?

FrameMaker and Word both save files in a proprietary format that allow creation of sophisticated page layouts that contain headers and footers, tables, bold headings, and so forth. Because you add Javadoc comments to Java source code files and then save them as text files, the only formatting features allowed in these files are HTML tags, such as `<bold> text </bold>` or `<code> text </code>`. As a result, you don't want all the formatting features that a word processor such as Word adds to files, often automatically.

Moreover, some features that are very helpful when writing code, such as *printing line numbers*, are advanced or non-existent features in Word and FrameMaker. These applications are designed for word processing and authoring long documents,

respectively, not for writing text files. But, you may ask, what about the text editor that comes with Windows? What about Notepad?

WHY NOT JUST USE NOTEPAD?

Notepad has a number of features that are annoying when you are working with *Java source files* (files that have a four-character java extension). For example, normally files are saved in Notepad with a three-character txt extension, although you can work around this by enclosing the file name in quotation marks (for example, “Pets.java”). If you forget to enclose the file name in quotation marks—an easy error to make when you are concentrating on the content of your files—the result is a file named Pets.java.txt. Notepad also requires you to have multiple instances of the application running, one instance for each open file. Lastly, Notepad does not have a keyboard shortcut (for example, CTRL-S) for saving a file, which is another annoyance when you are working in a text editor for an extended period of time and want to periodically save your work. Luckily, a number of reasonably priced alternatives are available.

FREWARE, SHAREWARE, AND COMMERCIAL TEXT EDITORS

All the text editors listed in this section have a graphical user interface (GUI), are easy to download and install, and offer keyboard shortcuts, macros, and other sophisticated text-manipulation features.

One handy way to categorize text editors is to divide them into freeware, shareware, and commercial text editors:

- Freeware, as the name implies, is free, and includes open source text editors. For more information about open source, go to <http://www.opensource.org>.
- Shareware can be used for a trial period, typically one month, before you must register and pay for it. Shareware is usually moderately priced.

Documenting APIs: Writing Developer Documentation for Java APIs and SDKs

- Commercial text editors tend to have better technical support and may have a trial evaluation period but are the most costly.

Freeware Text Editors

The following text editors are available for download free of charge:

- jEdit (available at <http://www.jedit.org/>). jEdit is open source, written in Java, and extensible. *Extensible* means, for example, that you can increase the functionality with a plug-in that adds features for editing HTML. Note that you have to install the JDK before you can use jEdit.
- Programmer's Notepad (available at <http://www.pnotepad.org>). Programmer's Notepad is an open-source text editor with a number of handy features, including syntax highlighting, line numbering, and a search-and-replace feature.
- Gnu Emacs (available at <http://www.gnu.org/software/emacs/emacs.html>). Emacs is open source, extensible, and has a dedicated user base. It also has extensive user documentation and a tutorial, features lacking in other freeware and shareware text editors.

Shareware Text Editors

Shareware offers a good compromise between freeware and commercial text editors although shareware text editors, unlike commercial text editors, typically lack a user manual. What they do often have, though, is an FAQ and other forms of online user support. Popular shareware text editors are:

- EditPlus (available at <http://www.editplus.com>). The registration fee is \$30. No user manual is available, although the web site does have screen captures, a user group, and an FAQ.
- TextPad (available at <http://www.textpad.com>). The registration fee is \$30. No user manual is available, but the web site offers an FAQ, a list of user-supplied tips, and a technical support forum.

- UltraEdit (available at <http://www.idmcomp.com>). The registration fee is \$30. No user manual is available, but the web site has support features that include a well-organized message board, screen captures, and a features list.

For the projects described in this book, you can use the latest version of any of these text editors.

Commercial Text Editors

Commercial text editors are expensive, and the authors find the features they offer do not warrant purchasing them, especially when compared to the available shareware text editors. For the sake of completeness, though, this chapter includes one.

Visual Slick Edit (available at <http://www.slickedit.com>) has a dedicated user base and the ability to emulate the UI of other popular editors, such as the Unix text editor, Vi. As this book goes to press, the cost is \$295.

FINAL THOUGHTS ABOUT CHOOSING A TEXT EDITOR

Choosing a text editor is ultimately a matter of personal preference, which you can determine by downloading a few text editors and using them. Choosing a few text editing tasks and seeing how each text editor handles them is an easy way to compare products. Most text editors have numerous useful features that take a long time to discover, but their shortcomings reveal themselves within a few days of moderate use.

The following features are the ones that the authors of this book consider *must-haves* in a text editor:

- Auto-indent
- Line numbering
- Macros
- Syntax color highlighting

Documenting APIs: Writing Developer Documentation for Java APIs and SDKs

- The ability to have more than one file open at a time

The following features are the ones that the authors of this book consider *nice-to-haves* in a text editor:

- Multi-file searching
- Multiple undo and redo
- Split screens, so you can view two areas of the same file simultaneously
- The ability to execute programs from within the editor

Downloading and Installing the JDK

The instructions in this section for downloading and installing the JDK 5.0 Update 7 provide a capsule summary. If you have trouble downloading the application or are unfamiliar with installing software, please see the detailed procedure we maintain on our web site at <http://www.bitzone.com>.

Note If you have a previous version of the Java SE Development Kit already installed, you do not need to uninstall it before you install the JDK 5.0 Update 7 unless you run out of space on your hard drive.

CAPSULE SUMMARY OF DOWNLOADING THE JDK 5.0 UPDATE 7

To download the JDK 5.0 Update 7:

- 1 Go to the Sun Microsystems Java J2SE site at:
<http://java.sun.com/javase/index.jsp>
- 2 Choose **Download**.

The title of the page is *Java SE Downloads*. The URL is:
<http://java.sun.com/javase/downloads/index.jsp>.

3 Choose the [Download](#) link for [JDK 5.0 Update 7](#).

Note that this page also contains links to [JDK 5.0 Update 7 with NetBeans 5.0](#), [Java Runtime Environment \(JRE\) 5.0 Update 7](#), and other very similar links that can be confusing. To clarify the correct choice, as of print date, the text for the [JDK 5.0 Update 7](#) download reads:

The J2SE Development Kit (JDK) includes:

- the Java Runtime Environment (JRE)
- command-line development tools, such as compilers and debuggers, that are necessary or useful for developing applets and applications

4 After accepting the license agreement, choose the [Windows offline installation](#) link.

The Windows platform download has two installation methods:

- The Windows *offline* installation choice downloads an executable file, which you then open to finish the installation.
- The Windows *online* installation choice downloads and installs directly from the Sun Microsystems Java web site.

The authors recommend that you select the first choice and download the executable file so that you have the flexibility to, for example, reinstall the JDK on a laptop without having to repeat the time-consuming download.

CAPSULE SUMMARY FOR INSTALLING THE [JDK 5.0 UPDATE 7](#)

To install the [JDK 5.0 Update 7](#):

- 1** Double-click the Java 5.0 executable file you downloaded from the Sun Microsystems web site.
- 2** Run the file.
- 3** Read and accept the license agreement, then choose **Next**.

- 4 Continue with the InstallShield Wizard, accepting the default installations for both the JDK and the JRE, until done.

Downloading and Installing Ant

This section covers how to download and install the Ant build tool.

CAPSULE SUMMARY FOR DOWNLOADING AND INSTALLING ANT

To download Ant:

- 1 Go to <http://ant.apache.org/>.
- 2 On the Welcome page, click the **download** link for the latest release of Ant. This procedure uses Ant 1.6.5 as an example.

You can also choose the **Binary Distributions** link located in the Download category. Click the Home tab to display the Categories list, if necessary.

- 3 Download the `apache-ant-1.6.5-bin.zip` archive file to an easily-located directory, such as one named `download`. Scroll down to see the file listed in the Current Release of Ant section, if necessary.

To install Ant:

- 1 Double-click the `apache-ant-1.6.5-bin.zip` archive file.
You can also choose **Open** in the Download Complete dialog.
- 2 When the file opens in Winzip, choose **Extract** and select your C drive. Ensure that you do not extract the files to the `download` directory but rather extract the files to a directory called `apache-ant-1.6.5`.

Setting the Environment Variables

As part of configuring your system, you need to set the `PATH` and `CLASSPATH` environment variables. This section covers creating a batch file to set them.

Introduction

The `PATH` environment variable indicates to the operating system which directories to look in for the executable files that run your applications. The operating system first looks in the current directory, then searches the drives and directories you have pointed to using the `PATH` environment variable. You need to add the JDK's `bin` directory to your existing path so that you have access to the JDK tools. (The `bin` directory is where executable files are stored.)

In addition, you need to set the `CLASSPATH` environment variable. The `CLASSPATH` environment variable allows the JDK tools to locate the Java API classes.

You can modify the environment variables using the Windows Start Menu > Control Panel. If you have never done this before, note that you can render your computer system unstable by improperly changing system variables.

To prevent trouble, if you are unfamiliar with modifying the environment variables, we suggest that you always use a batch file to set both the `PATH` and the `CLASSPATH` environment variables. With a batch file, each time you open a Command Prompt window and start a work session, you type `j15` (the name of the batch file) at the command prompt, and the batch file safely sets the variables. The following section provides detailed directions for creating this batch file.

Setting the PATH and CLASSPATH with a Batch File

To create a batch file called `j15.bat`:

- 1 Create a directory on your C drive called:

```
apidocs
```

- 2 Open a text editor and enter the following five lines in a file:

```
set JAVA_HOME=c:/Program Files/Java/jdk1.5.0_07
set JAVA_BIN=c:\Program Files\Java\jdk1.5.0_07\bin
set ANT_HOME=c:\apache-ant-1.6.5
set PATH=%JAVA_BIN%;%ANT_HOME%\bin;%PATH%
set CLASSPATH=.;%JAVA_HOME%\jre\lib\rt.jar
```

- 3 Save the file with the name `j15.bat` in the `C:\apidocs` directory.
- 4 Whenever you open a Command Prompt window and begin a work session, you immediately change to the `apidocs` directory by entering the following command at the command prompt (Figure 3.1):

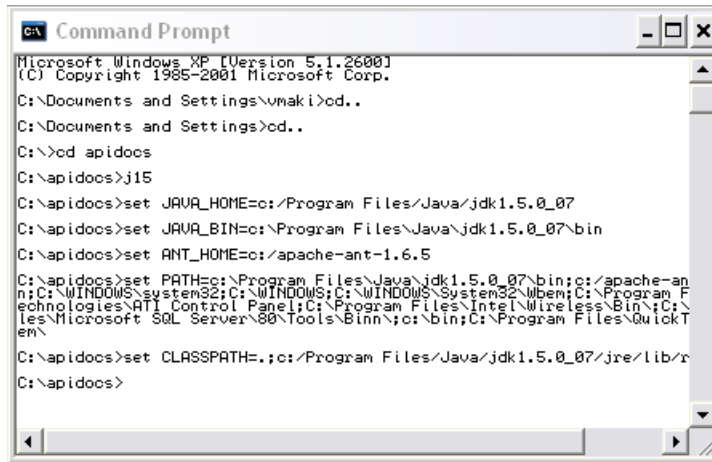
```
C:\>cd apidocs
```

- 5 At `C:\apidocs`, type:

```
j15
```

and press the Enter key.

You see your system being configured as shown in Figure 3.1. Your system is configured for the current work session only. If you close the Command Prompt window and reopen it, you must run the batch file again.

Figure 3.1 Screen display after running `j15.bat`

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\umaki>cd..
C:\Documents and Settings>cd..
C:\>cd apidocs
C:\apidocs>j15
C:\apidocs>set JAVA_HOME=c:/Program Files/Java/jdk1.5.0_07
C:\apidocs>set JAVA_BIN=c:\Program Files\Java\jdk1.5.0_07\bin
C:\apidocs>set ANT_HOME=c:/apache-ant-1.6.5
C:\apidocs>set PATH=c:\Program Files\Java\jdk1.5.0_07\bin;c:/apache-ant-1.6.5\bin;c:\WINDOWS\system32;c:\WINDOWS;c:\WINDOWS\System32\Wbem;c:\Program Files\ATI Technologies\ATI Control Panel;c:\Program Files\Intel\Wireless\Bin;c:\Program Files\Microsoft SQL Server\80\Tools\Binn;c:\bin;c:\Program Files\QuickTime\
C:\apidocs>set CLASSPATH=.;c:/Program Files/Java/jdk1.5.0_07/jre/lib/r
C:\apidocs>
```

Downloading the Sample Code

This section describes the sample code files you will need to complete the exercises in this book, and provides instructions for downloading and uncompressing (that is, *unzipping*) the sample code files on your hard drive.

Introduction

The `apidocs.zip` file contains the four sample Java code projects used in this book. Once you uncompress `apidocs.zip`, you see four directories:

- `pets`—Pets is a simple project with four classes used as example code to illustrate OOP concepts, such as classification, and Java basic syntax, such as method signatures.

Documenting APIs: Writing Developer Documentation for Java APIs and SDKs

- `hello`—The `HelloWorldWindow` project is the source project for the exercises in Chapter 6, “Java Practicum.” The exercises in Chapter 6 are divided into six phases so that you better understand how a program is typically developed and so that you practice compiling, running, and debugging code at each phase. The code for each phase is included in a subdirectory named respectively *01*, *02*, *03*, and so forth, for each phase of the exercises in Chapter 6.
- `taft`—The `taft` directory contains the source code for TAFT. TAFT is an example of a Java application that exposes a plug-in API.
- `taftPlugin`—The `taftPlugin` directory is an example of how a third-party developer would set up the build environment to write a plug-in for TAFT. This directory contains the runtime version of TAFT.

Requirements

To download and install the sample code projects in a Windows environment, you need the following programs:

- A web browser, such as Internet Explorer, Firefox, Mozilla, and so forth.
- A utility to uncompress ZIP files. If you have Windows XP, this functionality is built into the operating system. If you have an earlier version of Windows, you may need to install a utility program such as WinZip, which is available at <http://www.winzip.com>.

If you have a Macintosh, go to:

<http://www.bitzone.com/zone/abd>

for instructions on downloading and installing the sample code projects.

Downloading the Sample Code Projects

To download the `apidocs.zip` file:

- 1 Enter the following URL in a web browser:
`http://www.bitzone.com/zone/adb/apidocs.zip`
- 2 You will be prompted for a user id and password. Your user id and password is located on a sticker inside the front cover of this book.
- 3 Save the file to a directory on your hard drive, such as `download`.

Uncompressing the Sample Code Projects

You should uncompress the `apidocs.zip` file to a convenient directory on your hard drive. The instructions in this procedure suggest uncompressing the `zip` file to a top-level directory on your hard drive named `apidocs` (that is, `C:\apidocs`).

In Windows XP:

- 1 In the File Explorer, navigate to the directory where you downloaded the `apidocs.zip` file.
- 2 Double-click on the file icon. A window will open that contains four directories:
`hello`
`pets`
`taft`
`taftPlugin`
- 3 In the window, Shift-Select each of these directories and drag them to the `C:\apidocs` directory you previously created.
The sample files are copied to the `C:\apidocs` directory.

Documenting APIs: Writing Developer Documentation for Java APIs and SDKs