APPENDIX D

Installing the JDK and Using the JDK Documentation

To write and execute Java programs, you will need to download and install the Java Development Kit (JDK). For this text, you need the JDK Standard Edition, which is available for the Windows, Linux, macOS, and Solaris operating systems. It can be downloaded from this website:

```
www.oracle.com/technetwork/java/javase/downloads/index.html
```

NOTE: If you would rather not type this entire address into your browser, you can also go to `www.oracle.com`, click *Downloads*, then click *Java for Developers*.

On the webpage, click the link to download the JDK. You will need to accept the license agreement, then select the version of the JDK for your operating system to begin the download.

Once the file is downloaded, execute it to start the installation. The installation program will install two items to your system: the JDK and the Java Runtime Environment (JRE). First the JDK will be installed. Click the Next button on each screen to accept the default selections, and be sure to take note of the location on your system where the JDK will be installed. Depending on the version of the JDK that you are installing, on windows the location will be something similar to

```
C:\Program Files\Java\jdk-9.0.1
```

Once the JDK is installed, the installation process for the JRE begins. Click the Next button on each screen to accept the default selections, and be sure to take note of the location on your system where the JRE will be installed. Depending on the version that you are installing, on windows the location will be something similar to

```
C:\Program Files\Java\jre-9.0.1
```

Setting the `Path` Environment Variable

If you plan to use the JDK command-line utilities to compile and run your programs, you will probably want to edit the contents of the `Path` variable on your system. This allows your system to find the JDK utilities from any folder when you run them at the command-line.

The `Path` variable contains a list of directory paths separated by semicolons. For example, on windows the `Path` variable might contain something like the following:

```
C:\Games;C:\Temp;C:\Program Files\MyPrograms
```

When you type the name of an executable file at the command-line and press Enter ↵, the system will first look in the current folder for that file. If it cannot find the file there, it will begin looking in the folders that are listed in the `Path` variable. (On an actual system, the `Path` variable will contain many more paths than shown in this example, but this gives you an idea of how it works.)

We mentioned earlier that during the JDK installation process, you should take note of the location on your system where the JDK is installed. Inside that folder, there is another folder named `bin` that contains the JDK utility programs. The path to that folder on windows will be something like

```
C:\Program Files\Java\jdk-9.0.1\bin
```

> **NOTE:** Keep in mind that the actual path on your system might differ slightly from this example, depending on the version of the JDK that you have installed.

To make it easy to execute the JDK utilities from the Windows command line, you should add this path to the `Path` variable. The procedure for adding this path to the `Path` variable depends on the version of Windows you are using. The steps required for Windows 10, Windows 8, and Windows 7 follow.

Windows 10

In the Windows search box, type "environment variables" then select **Edit the system environment variables** from the search results.

This will display the *System Properties* window (with the *Advanced* tab selected at the top of the window). Click the Environment Variables ... button. Under **System Variables,** find *Path*, click on it, then click the Edit button. Add a semicolon to the end of the existing contents then add the path of the JDK utility programs. Click the OK buttons until all the dialog boxes are closed and exit the control panel.

Windows 8

In the right bottom corner of the screen, click on the **Search** icon and type **Control Panel.** Click on **Control Panel,** then click **System,** then click **Advanced system settings.** Click on the **Advanced** tab, then click Environment Variables. Under **System Variables,** find Path, click on it, then click the Edit button. Add a semicolon to the end of the existing contents and then add the path of the JDK utility programs. Click the OK buttons until all the dialog boxes are closed and exit the control panel.

Windows 7

Click the Start button then right-click **Computer**. On the pop-up menu, select **Properties**. In the window that appears next, click **Advanced system settings**. This displays the System Properties window. Click the Environment Variables ... button. In the System Variables list, scroll to the `Path` variable. Select the Path variable and click the Edit button. Add a semicolon to the end of the existing contents, and then add the path of the JDK utility programs. Click the OK buttons until all the dialog boxes are closed and exit the control panel.

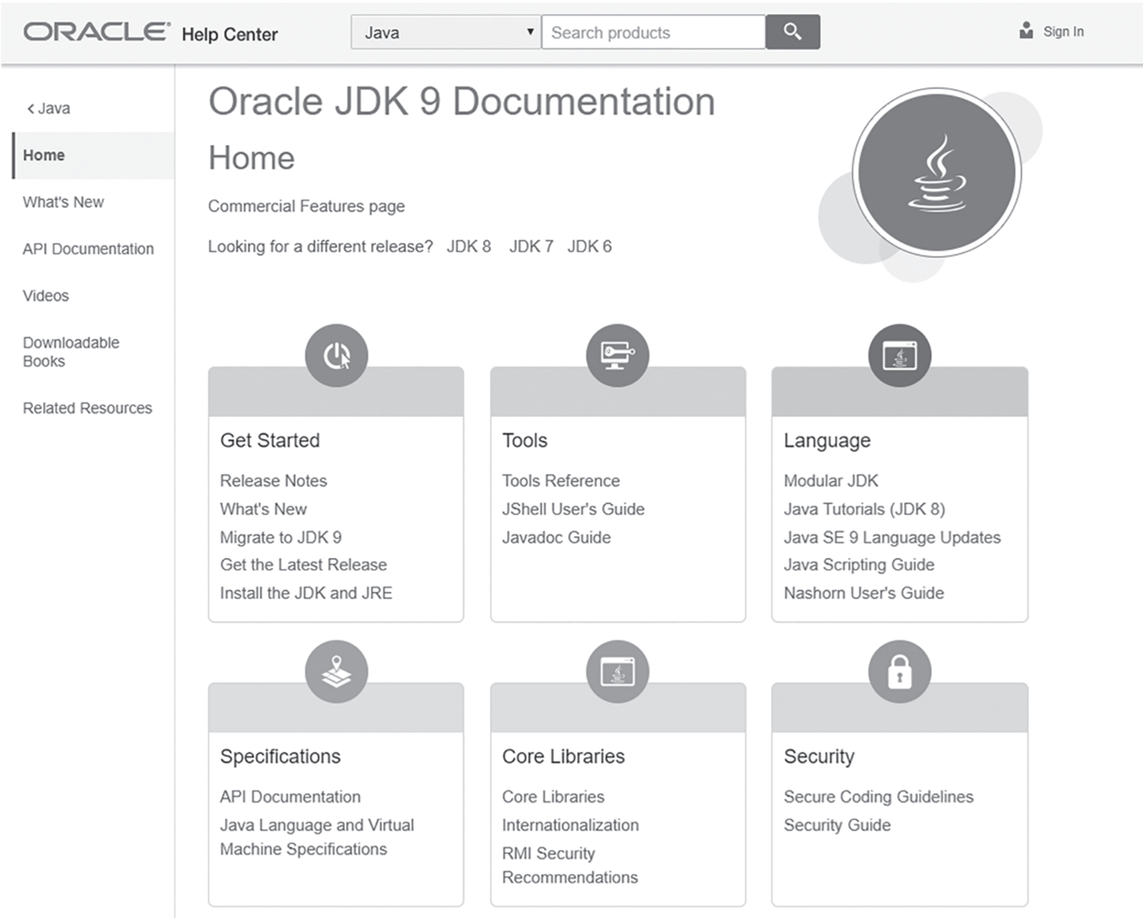
Using the Java API Documentation

To view the official Java API documentation, go to the following site:

```
http://docs.oracle.com/javase
```

Figure D-1 shows how the page appears at the time this was written. The page provides links to several useful sites, including the official *API documentation*. As a student learning Java, or as a professional Java programmer, you will find yourself referring to the API documentation frequently.

Figure D-1 The Java SE documentation site (Oracle Corporate Counsel)



Oracle JDK 9 Documentation

Home

Commercial Features page

Looking for a different release? JDK 8 JDK 7 JDK 6

Get Started
- Release Notes
- What's New
- Migrate to JDK 9
- Get the Latest Release
- Install the JDK and JRE

Tools
- Tools Reference
- JShell User's Guide
- Javadoc Guide

Language
- Modular JDK
- Java Tutorials (JDK 8)
- Java SE 9 Language Updates
- Java Scripting Guide
- Nashorn User's Guide

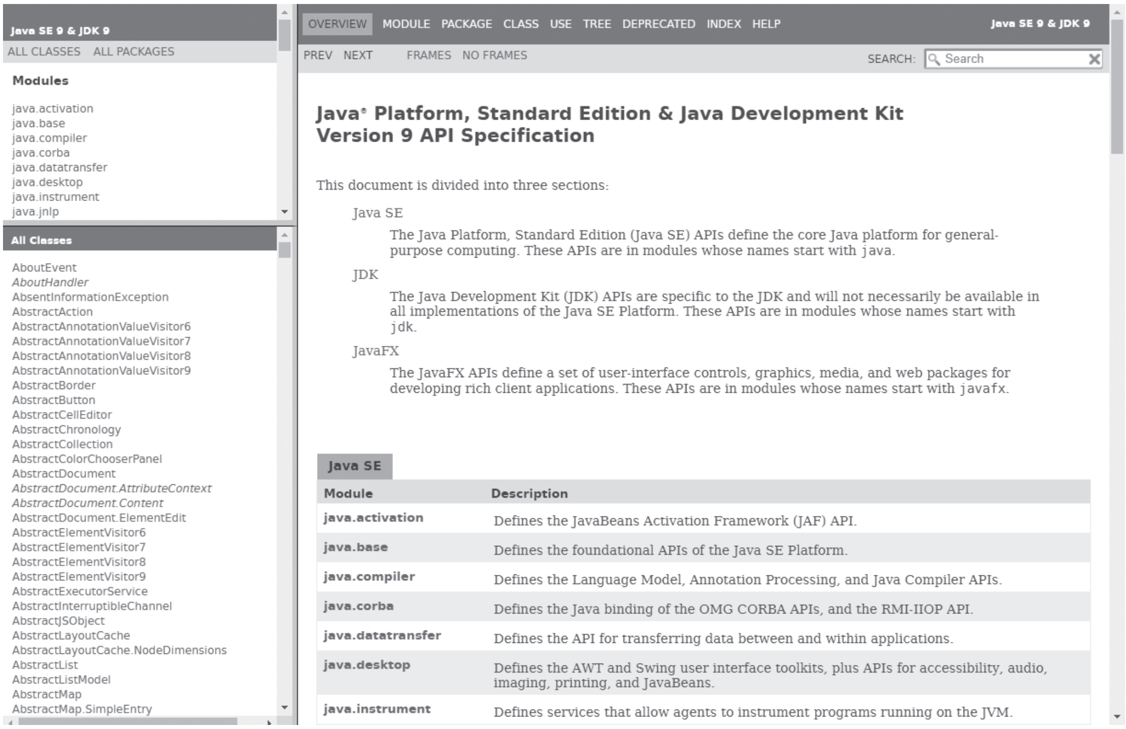
Specifications
- API Documentation
- Java Language and Virtual Machine Specifications

Core Libraries
- Core Libraries
- Internationalization
- RMI Security Recommendations

Security
- Secure Coding Guidelines
- Security Guide

When you click the link for the *API documentation*, and then click the link for *Frames*, you will see a page similar to the one shown in Figure D-2. In the leftmost pane, you will see an alphabetically ordered list of all the classes and interfaces in the Java API. To see the documentation for a particular class, simply scroll to the name of the class in the left pane, and click it.

Figure D-2 Java SE API documentation (Oracle Corporate Counsel)



Modules

java.activation
java.base
java.compiler
java.corba
java.datatransfer
java.desktop
java.instrument
java.jnlp

All Classes

AboutEvent
AboutHandler
AbsentInformationException
AbstractAction
AbstractAnnotationValueVisitor6
AbstractAnnotationValueVisitor7
AbstractAnnotationValueVisitor8
AbstractAnnotationValueVisitor9
AbstractBorder
AbstractButton
AbstractCellEditor
AbstractChronology
AbstractCollection
AbstractColorChooserPanel
AbstractDocument
AbstractDocument.AttributeContext
AbstractDocument.Content
AbstractDocument.ElementEdit
AbstractElementVisitor6
AbstractElementVisitor7
AbstractElementVisitor8
AbstractElementVisitor9
AbstractExecutorService
AbstractInterruptibleChannel
AbstractJSObject
AbstractLayoutCache
AbstractLayoutCache.NodeDimensions
AbstractList
AbstractListModel
AbstractMap
AbstractMap.SimpleEntry

Java® Platform, Standard Edition & Java Development Kit Version 9 API Specification

This document is divided into three sections:

Java SE

The Java Platform, Standard Edition (Java SE) APIs define the core Java platform for general-purpose computing. These APIs are in modules whose names start with `java`.

JDK

The Java Development Kit (JDK) APIs are specific to the JDK and will not necessarily be available in all implementations of the Java SE Platform. These APIs are in modules whose names start with `jdk`.

JavaFX

The JavaFX APIs define a set of user-interface controls, graphics, media, and web packages for developing rich client applications. These APIs are in modules whose names start with `javafx`.

Java SE

Module	Description
java.activation	Defines the JavaBeans Activation Framework (JAF) API.
java.base	Defines the foundational APIs of the Java SE Platform.
java.compiler	Defines the Language Model, Annotation Processing, and Java Compiler APIs.
java.corba	Defines the Java binding of the OMG CORBA APIs, and the RMI-IIOP API.
java.datatransfer	Defines the API for transferring data between and within applications.
java.desktop	Defines the AWT and Swing user interface toolkits, plus APIs for accessibility, audio, imaging, printing, and JavaBeans.
java.instrument	Defines services that allow agents to instrument programs running on the JVM.