Appendix

Reading and Understanding Java's API Documentation

Before Java was born, people judged programming languages solely by their structural features. Does an `if` statement do what you expect it to do? Are looping statements easy to use? Are methods implemented efficiently?

With Java, things are a bit different. Sure, Java has a whole collection of built-in language features. But Java is much more than just a big set of grammar rules. Java has a standard Application Programming Interface -- a huge library consisting of at least 3,000 canned programs, each with its own functionality, its own limitations, and its own rules for effective use.

How do you figure out how to use all these programs? The answer is, you don't. You figure out how to use a few, and you read Java's API documentation. With this documentation, you can find information you need, when you need it.

Searching for a Term

You can find things in the API documentation in a number of different ways. Each way is convenient in one situation or another. For instance, in many of this book's listings, I call a method named `System.out.println`. The rest of this appendix describes two ways to look up the `System.out.println` method.

Using the index

Here's how to find something, such as `System.out.println`, by using the index:

1. **Download Sun's Java API documentation.**

   For more help on downloading the documentation, see Chapter 2.

2. **Open to the front page of the documentation.**

When you download the documentation, you get several directories. In the top-level directory is a file named `index.html` (or `index.htm`). Open this file in your Web browser.

3. **Click the API & Language link, which is near the top of the front page, as shown in Figure A-1.**

   This takes you farther down on the same Web page.

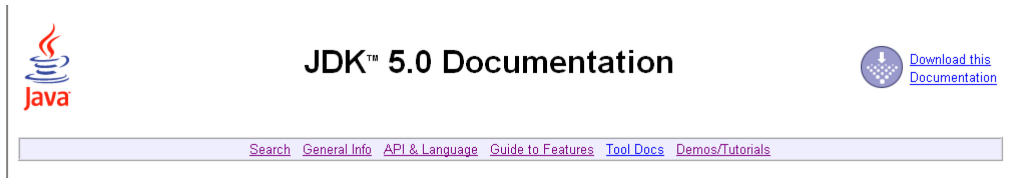


Figure A-1: The front page of Sun's documentation.

4. **Click the Java 2 Platform API Specification link, as shown in Figure A-2.**

   The browser transports you to the start of the API pages, which are shown in Figure A-3.

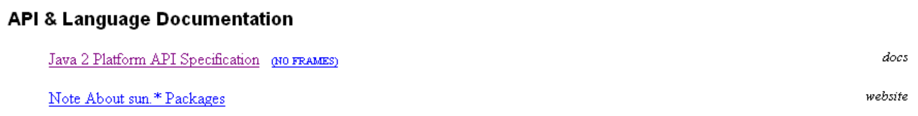
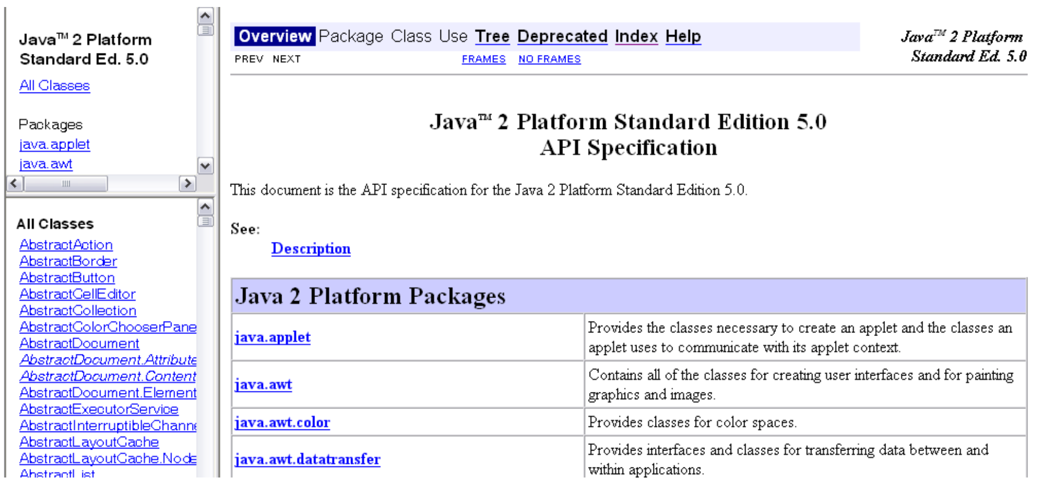

API & Language Documentation
Java 2 Platform API Specification
(NO FRAMES)
docs
Note About sun.* Packages
website

Figure A-2: A link to the API specification.



Java™ 2 Platform Standard Ed. 5.0

All Classes

Packages
java.applet
java.awt

All Classes
AbstractAction
AbstractBorder
AbstractButton
AbstractCellEditor
AbstractCollection
AbstractColorChooserPane
AbstractDocument
AbstractDocument.Attribute
AbstractDocument.Content
AbstractDocument.Element
AbstractExecutorService
AbstractInterruptibleChann
AbstractLayoutCache
AbstractLayoutCache.Node

Overview Package Class Use **Tree** **Deprecated** **Index** **Help**
PREV NEXT FRAMES NO FRAMES

Java™ 2 Platform Standard Ed. 5.0

Java™ 2 Platform Standard Edition 5.0
API Specification

This document is the API specification for the Java 2 Platform Standard Edition 5.0.

See:
Description

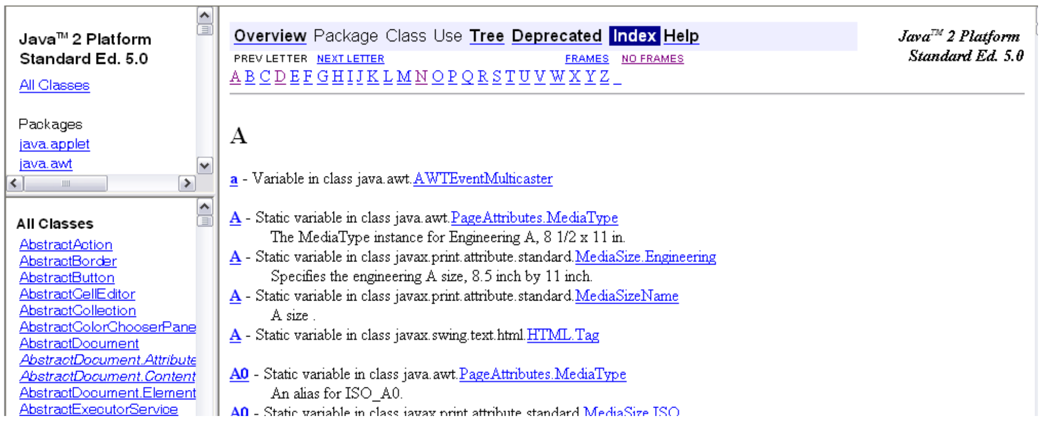
Java 2 Platform Packages	
java.applet	Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.
java.awt	Contains all of the classes for creating user interfaces and for painting graphics and images.
java.awt.color	Provides classes for color spaces.
java.awt.datatransfer	Provides interfaces and classes for transferring data between and within applications.

Figure A-3: The start of the documentation's API pages.

Copyright 2004 Sun Microsystems, Inc. Reprinted with permission.

5. **Click the Index link at the top of the page to open the index, as shown in Figure A-4.**

   A list of letters is near the top of the index. Click the P link to go to the section with `println` in it.



Java™ 2 Platform Standard Ed. 5.0

All Classes

Packages
java.applet
java.awt

All Classes
AbstractAction
AbstractBorder
AbstractButton
AbstractCellEditor
AbstractCollection
AbstractColorChooserPane
AbstractDocument
AbstractDocument.Attribute
AbstractDocument.Content
AbstractDocument.Element
AbstractExecutorService

Overview Package Class Use **Tree** **Deprecated** **Index** **Help**
PREV LETTER NEXT LETTER FRAMES NO FRAMES
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z _

Java™ 2 Platform Standard Ed. 5.0

A

a - Variable in class java.awt.AWTEventMulticaster

A - Static variable in class java.awt.PageAttributes.MediaType
The MediaType instance for Engineering A, 8 1/2 x 11 in.
A - Static variable in class javax.print.attribute.standard.MediaSize.Engineering
Specifies the engineering A size, 8.5 inch by 11 inch.
A - Static variable in class javax.print.attribute.standard.MediaSizeName
A size .
A - Static variable in class javax.swing.text.html.HTML.Tag

A0 - Static variable in class java.awt.PageAttributes.MediaType
An alias for ISO_A0.
A0 - Static variable in class javax.print.attribute.standard.MediaSize.ISO

Figure A-4: The API documentation's index.

Copyright 2004 Sun Microsystems, Inc. Reprinted with permission.

6. **In the P section, do a search for `println` to find the `println` entries.**

   Most Web browsers enable you to search for something like `println` in the text of a page. Here's how:

A. Make sure the browser knows that you want to search in the big frame that takes up most of the page (and not in the smaller frames on the left side of the page). To do this, click your mouse inside the big frame. (Don't click a link. Click on some neutral white area of the frame.)

B. Open the browser's Find dialog box. On most browsers, pressing Ctrl+F coaxes the Find dialog box out of hiding.

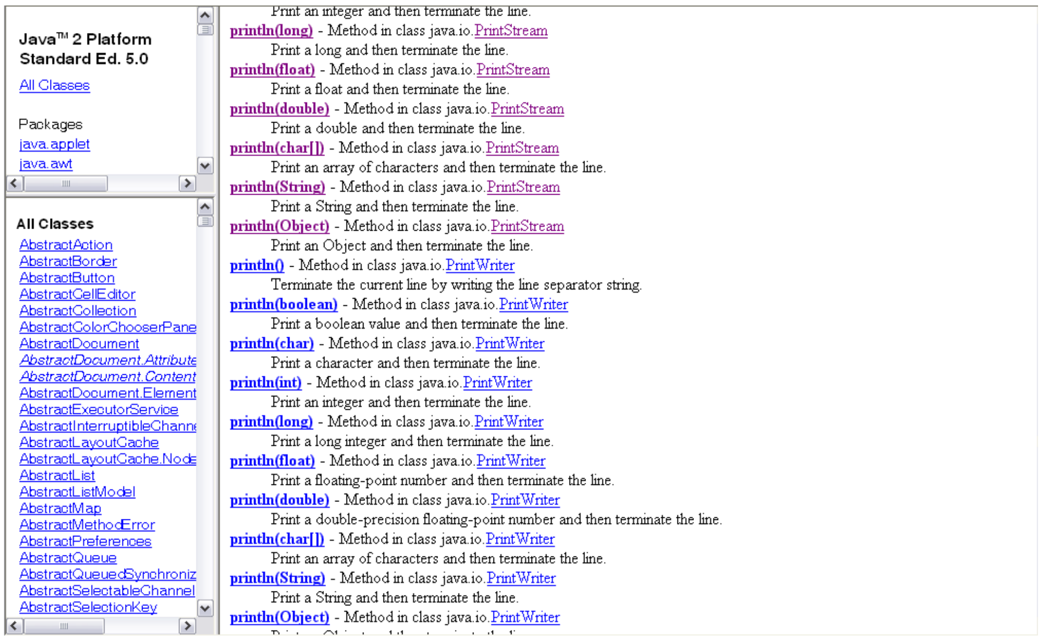
C. When you see the Find dialog box, type **println** in the text box and click the box's Find or Find Next button.

7. **Pick one of the `println` entries.**

   The P section has a big boatload of `println` entries, as shown in Figure A-5. The entries differ from one another in two ways:

   * Each entry says `println(int)`, `println(String)`, or `println(`*`someOtherTypeName`*`)`. The type name can differ from one entry to another.

   * Each entry says that `println` is a method in class `java.`*`someStuff.someMoreStuff`*. The class can differ from one entry to another.



Java™ 2 Platform
Standard Ed. 5.0

All Classes

Packages
java.applet
java.awt

All Classes
AbstractAction
AbstractBorder
AbstractButton
AbstractCellEditor
AbstractCollection
AbstractColorChooserPane
AbstractDocument
AbstractDocument.Attribute
AbstractDocument.Content
AbstractDocument.Element
AbstractExecutorService
AbstractInterruptibleChann
AbstractLayoutCache
AbstractLayoutCache.Node
AbstractList
AbstractListModel
AbstractMap
AbstractMethodError
AbstractPreferences
AbstractQueue
AbstractQueuedSynchroniz
AbstractSelectableChannel
AbstractSelectionKey

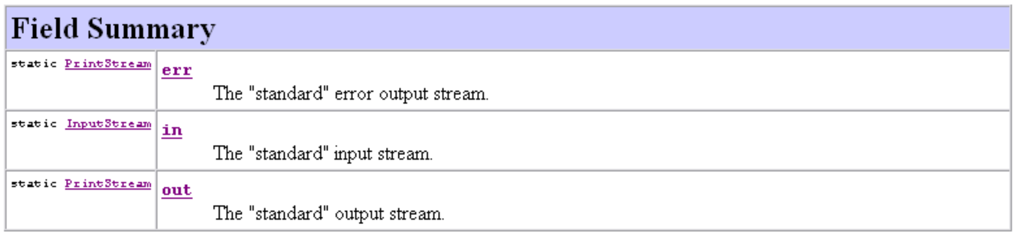
Print an integer and then terminate the line.
println(long) - Method in class java.io.PrintStream
Print a long and then terminate the line.
println(float) - Method in class java.io.PrintStream
Print a float and then terminate the line.
println(double) - Method in class java.io.PrintStream
Print a double and then terminate the line.
println(char[]) - Method in class java.io.PrintStream
Print an array of characters and then terminate the line.
println(String) - Method in class java.io.PrintStream
Print a String and then terminate the line.
println(Object) - Method in class java.io.PrintStream
Print an Object and then terminate the line.
println() - Method in class java.io.PrintWriter
Terminate the current line by writing the line separator string.
println(boolean) - Method in class java.io.PrintWriter
Print a boolean value and then terminate the line.
println(char) - Method in class java.io.PrintWriter
Print a character and then terminate the line.
println(int) - Method in class java.io.PrintWriter
Print an integer and then terminate the line.
println(long) - Method in class java.io.PrintWriter
Print a long integer and then terminate the line.
println(float) - Method in class java.io.PrintWriter
Print a floating-point number and then terminate the line.
println(double) - Method in class java.io.PrintWriter
Print a double-precision floating-point number and then terminate the line.
println(char[]) - Method in class java.io.PrintWriter
Print an array of characters and then terminate the line.
println(String) - Method in class java.io.PrintWriter
Print a String and then terminate the line.
println(Object) - Method in class java.io.PrintWriter

Figure A-5: Some println entries in the API documentation's index.

At this point, it pays to poke around. If you're trying to print something like `"Hello world!"`, you want one of the `println(String)` entries. On the other hand, if you're trying to print the value of `amountInAccount`, you'll probably choose a `println(double)` entry.

Now, suppose you've decided on `println(String)`. You can choose from three `println(String)` entries. One says it's a method in class `java.io.PrintStream`, the next is a method in class `java.io.PrintWriter`, and the third is a method in class `java.sql.DriverManager`. Which of these three entries do you choose?

Well, what you're really trying to call is something named `System.out.println`. If you go through the whole lookup rigmarole with `System.out`, you'll find that `System.out` has type `PrintStream`. (See Figure A-6.) So the entry you decide to choose is `println(String) - Method in class java.io.PrintStream`.



Field Summary

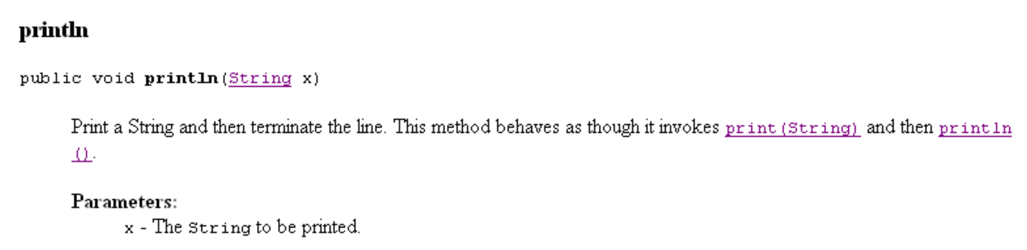
static PrintStream	**err** The "standard" error output stream.
static InputStream	**in** The "standard" input stream.
static PrintStream	**out** The "standard" output stream.

Figure A-6: The out variable has type PrintStream.

Copyright 2004 Sun Microsystems, Inc. Reprinted with permission.

8. **Click the link for the entry that you've chosen.**

   When you click the println(String) link, the browser takes you to a page that explains a `println` method, as shown in Figure A-7. The page tells you what `println` does ("Print a String and then. . . .") and points to other useful pages, like the page with the documentation for `String`.



println

```
public void println(String x)
```

Print a String and then terminate the line. This method behaves as though it invokes `print(String)` and then `println()`.

Parameters:
`x` - The `String` to be printed.

Figure A-7: A description of the println method.

Copyright 2004 Sun Microsystems, Inc. Reprinted with permission.

Using the list of classes

Here's how to find an entry in the API by starting in the list of classes:

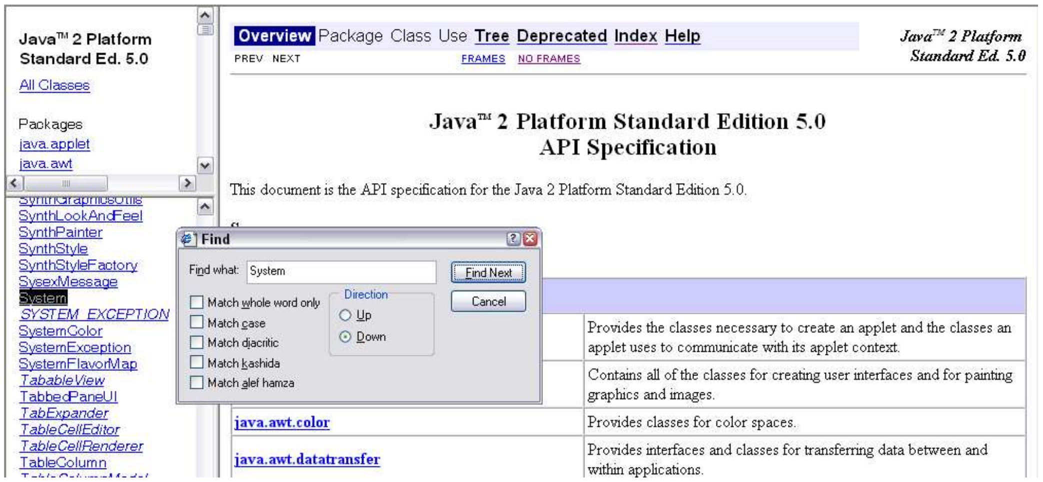
1. **Navigate to the start of the documentation's API pages.**

   To do this, follow the first four steps in this appendix's "Using the index" section.

2. **Find the page that documents the `System` class.**

   You're looking for documentation that explains `System.out.println`. So you look up `System`, work your way to `out`, and from there, work your way to `println`.

   To find a link to `System`, look in the lower frame on the left side of the page. (See Figure A-8.) For hints on finding text on the page, see Step 6 in the "Using the index" section.



Java™ 2 Platform Standard Edition 5.0
API Specification

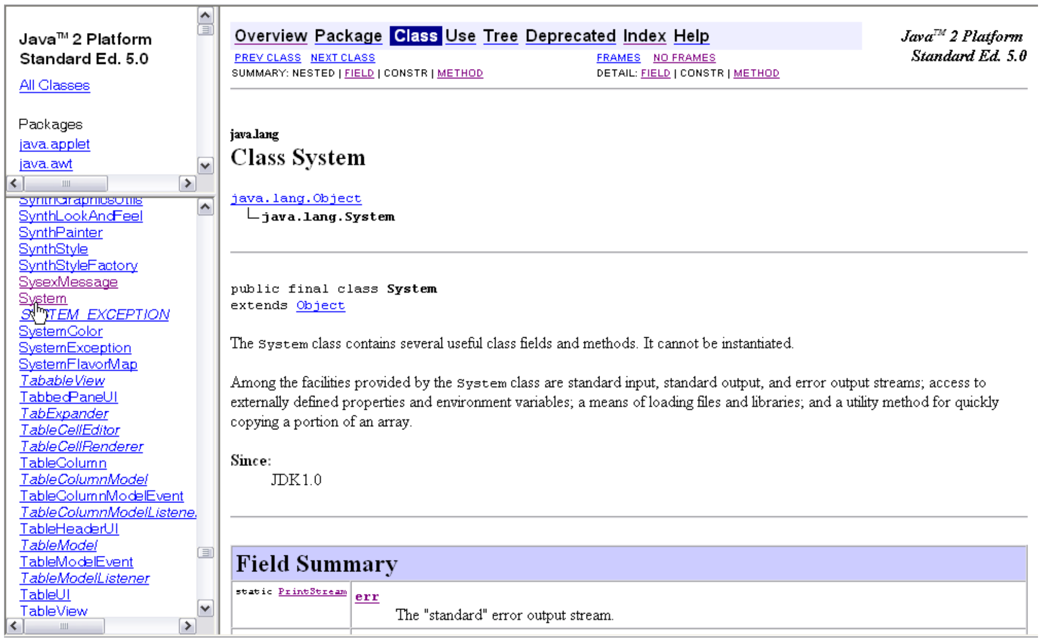
This document is the API specification for the Java 2 Platform Standard Edition 5.0.

	Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.
	Contains all of the classes for creating user interfaces and for painting graphics and images.
java.awt.color	Provides classes for color spaces.
java.awt.datatransfer	Provides interfaces and classes for transferring data between and within applications.

Figure A-8: Finding a link to the System class.

Copyright 2004 Sun Microsystems, Inc. Reprinted with permission.

Clicking the System link makes your browser display the documentation page for the `System` class, as shown in Figure A-9.



java.lang
Class System

```
java.lang.Object
  └java.lang.System
```

```
public final class System
extends Object
```

The `System` class contains several useful class fields and methods. It cannot be instantiated.

Among the facilities provided by the `System` class are standard input, standard output, and error output streams; access to externally defined properties and environment variables; a means of loading files and libraries; and a utility method for quickly copying a portion of an array.

Since:
JDK1.0

Field Summary

static PrintStream	err The "standard" error output stream.

Figure A-9: The System class's documentation.

Copyright 2004 Sun Microsystems, Inc. Reprinted with permission.

3. **On the documentation page for the `System` class, find the `out` variable.**

   If you use your Web browser's Find dialog box, you have to click the Find Next button several times. (The word "out" is so common, it appears several times in several different contexts on the `System` documentation page.) When you've found what you're looking for, you see a table like the one shown in Figure A-6.

4. **In the table's `out` row, click the PrintStream link.**

   According to the documentation, the `out` variable refers to an object of type `PrintStream`. This means that `println` is part of the `PrintStream` class. That's why you're clicking the `PrintStream` link.

5. **On the documentation page for `PrintStream`, find `println(String)`.**

   You see an explanation like the one shown in Figure A-7.

You Can Do It Too

After following the steps in this appendix, you may be tempted to say, "Big deal, I can find `println` in the API docs, but I probably can't find anything else. And if people create documentation for stuff that they program on their own, then their documentation won't look like the standard API documentation. I'll be up a creek."

My response to all this is "Nonsense!" Here's why:

* Most of the tricks you need for finding things in the standard Java documentation are illustrated in this appendix's step-by-step instructions. If you can find `System.out.println`, you can also find `javax.swing.JButton` or any of the 3,000 programs in the standard Java API.

  And, as you discover more about Java and the relationships among classes, methods, and variables, this appendix's step-by-step instructions will feel much more natural.

* As for reading other people's documentation, you can scratch that problem right off your list. The standard API docs weren't typed by hand. They were generated automatically from actual Java program code. For instance, the code for `PrintStream.java` has a few lines that look something like this:

```
/**
 * Print a String and then terminate the line.
 * This method behaves as though it invokes
 * <code>{@link #print(String)}</code>
 * and then <code>{@link #println()}</code>.
 *
 * @param x  The <code>String</code> to be printed.
 */
```

To create the API documentation, the folks from Sun Microsystems ran a program called *javadoc*. The `javadoc` program took lines like these right out of the `PrintStream.java` file and used the lines to make the documentation that you see in your Web browser.

Other Java programmers `--` people who don't work for Sun Microsystems `--` do the same thing. In fact, everyone who writes Java code uses the `javadoc` program to generate documentation. So everyone's Java documentation looks like everyone else's Java documentation. When you know how read to the standard API documentation, you know how to read anybody's homegrown Java docs.

And yes, you can use the `javadoc` program too. When you download the JDK (see Chapter 2 for the details), you get the `javadoc` program as part of the deal. To find out more about turning your program comments into Web pages, visit this book's Web site.