

The Java™ Platform
A White Paper

Douglas Kramer

With contributions by
Bill Joy and David Spohnhoff



2550 Garcia Avenue
Mountain View, CA 94043 U.S.A.
408-343-1400
May 1996

Copyright Information

© 1995, 1996, Sun Microsystems, Inc. All rights reserved.
2550 Garcia Avenue, Mountain View, California 94043-1100 U.S.A.

This document is protected by copyright. No part of this document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

The information described in this document may be protected by one or more U.S. patents, foreign patents, or pending applications.

TRADEMARKS

Sun, Sun Microsystems, Sun Microelectronics, the Sun Logo, SunXTL, JavaSoft, JavaOS, the JavaSoft Logo, Java, HotJava, JavaChips, picoJava, microJava, UltraJava, JDBC, the Java Cup and Steam Logo, "Write Once, Run Anywhere" and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

UNIX[®] is a registered trademark in the United States and other countries, exclusively licensed through X/Open Company, Ltd.

Adobe[®] is a registered trademark of Adobe Systems, Inc.

Netscape Navigator[™] is a trademark of Netscape Communications Corporation.

All other product names mentioned herein are the trademarks of their respective owners.

THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THE DOCUMENT. SUN MICROSYSTEMS, INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.



Please
Recycle

The Java™ Platform

What is the Java Platform?	6
The Java Base Platform	7
The Embedded Java Platform	7
Benefits of the Java Platform	8
Applets and Applications	9
Where Will the Java Platform Be Deployed?	10
JavaChip™ Family	11
JavaOS™	12
A Word About the Java Language	12
A Look Inside the Java Platform	14
Java Virtual Machine	15
Java Base API	16
Java Applet API	16
Java Standard Extension API	16
Java Security API	17
Java Media API	18
Java Enterprise API	19
Java Commerce API	20
Java Server API	21
Java Management API	22
Java Compile and Runtime Environments	24



The Java™ Platform



This paper defines the Java™ Platform and provides descriptions of each of its parts. It is written for developers and others interested in understanding the wide range of environments where Java-powered applets and applications can run.

For more up-to-date and detailed information about the features and architecture of the Java Platform and development environment, refer to the Developer's Corner at the JavaSoft web site <http://java.sun.com>. In particular, see *The Java Language Environment: A White Paper*.

The Java Platform is developed by JavaSoft, an operating company of Sun Microsystems, Inc.



What is the Java Platform?

The computer world currently has many platforms, among them Microsoft Windows, Macintosh, OS/2, UNIX® and NetWare®; software must be compiled separately to run on each platform. The binary file for an application that runs on one platform cannot run on another platform, because the binary file is machine-specific.

The Java Platform is a new software platform for delivering and running highly interactive, dynamic, and secure applets and applications on networked computer systems. But what sets the Java Platform apart is that it sits on top of these other platforms, and compiles to *bytecodes*, which are not specific to any physical machine, but are machine instructions for a *virtual machine*. A program written in the Java Language compiles to a bytecode file that can run wherever the Java Platform is present, on *any* underlying operating system. In other words, the same exact file can run on any operating system that is running the Java Platform. This portability is possible because at the core of the Java Platform is the Java Virtual Machine.

While each underlying platform has its own implementation of the Java Virtual Machine, there is only one virtual machine specification. Because of this, the Java Platform can provide a standard, uniform programming interface to applets and applications on any hardware. The Java Platform is therefore ideal for the Internet, where one program should be capable of running on any computer in the world. The Java Platform is designed to provide this “Write Once, Run Anywhere”SM capability.

Developers use the Java Language to write source code for Java-powered applications. They compile once to the Java Platform, rather than to the underlying system. Java Language source code compiles to an intermediate, portable form of bytecodes that will run anywhere the Java Platform is present.

Developers can write object-oriented, multithreaded, dynamically linked applications using the Java Language. The platform has built-in security, exception handling, and automatic garbage collection. Just-in-time compilers are available to speed up execution by converting Java bytecodes into machine language. From within the Java Language, developers can also write and call native methods—methods in C, C++ or another language, compiled to a specific underlying operating system—for speed or special functionality.



The Java Language is the the entry ramp to the Java Platform. Programs written in the Java Language and then compiled will run on the Java Platform. The Java Platform has two basic parts:

- Java Virtual Machine
- Java Application Programming Interface (Java API)

These are described in detail later in this paper. Combined, these parts provide an end-user runtime environment for deploying Internet and intranet applications.

The Java Base Platform

The Java Base Platform is the *minimum* Java Platform that developers can safely assume is present for running Java-powered applets and applications. This platform applies to Network Computers, desktop computers, and workstations (the next section describes the platform for smaller systems). This platform contains the same Java Virtual Machine mentioned before, but has a minimal set of API required to run basic applets and applications. This minimal set is known as the Java Applet API, or Java Base API. Developers who write to this minimum set can feel secure that the program will run anywhere without the need for additional class libraries.

Certain Java Platform licensees (listed in “Where Will the Java Platform Be Deployed?” on page 10”) have contracted to include the Java Base API in their particular implementation of the Java Platform. As more class libraries are developed, the Java Base Platform will grow, and these additions will migrate in a timely fashion into the Java Base Platform present on each licensee’s operating system.

Another set of APIs, called the Standard Extension API, is being defined by JavaSoft, in partnership with leading industry companies, to extend the base functionality. Over time, some subset of the Standard Extension API will migrate into the Java Base Platform.

The Embedded Java Platform

The Embedded Java Platform is being targeted for consumer devices with fewer resources and more specialized functionality than a Network Computer, such as set-top boxes, printers, copiers, and cellular phones. Such devices might have special constraints such as small memory footprint, no display, or no connection to a network.



The API targeted for this platform is called the Java Embedded API. The Java Embedded API is the smallest API a low-function embedded device can have and still run. Because this platform is still under development, this API has not yet achieved the level of a standard. Consequently this API is not yet well-defined, but it will probably consist of the packages `java.lang` and `java.util`. A Java-powered application written for one particular device could operate on a wide range of similar, dedicated devices.

Benefits of the Java Platform

The Java Platform has benefits for the end-user as well as the developer and support personnel:

End-User Benefits

Today, the Java Platform provides live, interactive content on the World Wide Web, with just-in-time software access. Applications are readily available on all operating systems at once, freeing users from having to choose operating systems on that basis. Smaller, less expensive, dedicated systems will eventually be available for specialized applications.

Developer Benefits

The Java Language is a small, “knowable” system and is coupled with a growingly comprehensive set of APIs. Developers can “Write Once, Run Anywhere,” which provides tremendous marketing leverage over other languages. In addition, Java development environments on all operating systems compile to a single binary format. Rather than developing on multiple platforms to deliver on multiple platforms, developers can now develop on one platform, saving cost, to deliver on that same platform, which is everywhere. The ability to “Write Once, Run Anywhere” is enough reason for some developers to turn to the Java Language as an alternative to C or C++ even for stand-alone, non-networked applications.

In addition, building applications from shared, reusable objects can further reduce cost by allowing developers to concentrate on creating only what is novel. Developers can distribute by network rather than compete for shelf-space in software stores.



Administrative and Support Benefits

The Java Platform has benefits for corporate computer systems administration departments. Version control and upgrades are simplified because Java-powered applications can be kept in a central repository and served from there for each individual use. In multivendor, multiplatform environments, the number of platforms to support is reduced to one. Emerging lower-cost network computers have the potential to reduce maintenance and capital expenditures. With these network computers, data management can remain centralized while data processing is done locally.

Companies with large intranets, that may not find it worthwhile to upgrade to the latest memory-consuming operating system, can run Java-powered applications on all their existing machines. By providing corporate data in a format readable by Java-powered applications, corporations give users the platform-neutral access to the data they need.

When customers are running on the Java Platform, companies can take advantage of the interactivity of the Internet by moving employee tasks out to customers. Companies can reduce time spent on order-entry by having customers fill in order-entry forms themselves on Web pages. This is more practical than previously possible, because the customer can now be on any operating system.

Applets and Applications

The Java Platform enables developers to create two different kinds of programs:

- **Applets** are programs that require a browser to run. The `<applet>` tag is embedded in a Web page and names the program to be run. When that page is accessed by a user, either over the Internet or corporate intranet, the applet automatically downloads from the server and runs on the client machine. Because applets are downloaded, they tend to be designed small or modular, to avoid large download times.
- **Applications** are programs that do not require a browser to run—they have no built-in downloading mechanism. When an application is called, it runs. In this way, applications are just like programs in other languages. They can perform traditional desktop tasks, such as that done with a word processor, spreadsheet or graphics application. Like an applet, an application requires the Java Platform for it to run; however the Platform can be available as a



separate program, can be embedded directly within the underlying operating system, or presumably can be embedded in the application itself.

While applets and applications have different means of being invoked, for the most part they have the same access to a wide range of language capabilities. For example, either an applet or an application can access a host database, retrieve the data it needs, do local data processing, and store the results back to the host.

However, an applet requires a network to run, while an application does not. Applications have greater freedom in that they have full access to system services. For example an application, unlike an applet, can have normal read and write access to files on any disk. Since an applet can potentially be downloaded from an untrusted Web page, it is restricted from having read or write access to any file system except the server from which it came. This constraint will be relaxed when applets can be marked with digital signatures, allowing the end-user to be assured that it has been downloaded unaltered from a trusted source. Where local file storage is required, currently an application is required.

Where Will the Java Platform Be Deployed?

The progression towards ubiquity has great momentum, moving in three stages from browsers, to desktop, workstation and network operating systems, and finally to embedded devices.

First, the Java Base Platform is currently embedded in the most widely used Internet browser, Netscape Navigator™, and will soon be in Microsoft Internet Explorer. It is, or will become, available in other browsers, such as HotJava™.

Second, the Java Base Platform will soon be embedded in all leading desktop, workstation, and network operating systems—see Figure 1. Being available on the combination of Microsoft Windows, Macintosh, OS/2, and UNIX computers, the Java Base Platform will have an installed base as large as these platforms combined. By targeting this platform, developers will tap into the new, exploding market for Web and intranet applications without being tied to any particular hardware or operating system environment. The Java Platform will become the platform for all network- and Web-based computing.

Third, with the JavaChip™ family of integrated circuits, the platform will be available in a wide range of consumer and industrial embedded devices such as dedicated Network Computers, set-top boxes, printers, copiers and cellular phones.



Operating Systems That Embed the Java Base Platform	
Windows	
• Microsoft Corporation	Windows 95, Windows NT
• International Business Machines	Windows 3.1
Macintosh	
• Apple Computer, Inc.	MacOS
OS/2	
• International Business Machines	OS/2
Unix	
• Hewlett Packard Corp.	HPUX
• Hitachi, Ltd.	Hitachi OS
• International Business Machines	AIX
• Silicon Graphics, Inc.	Irix
• SunSoft, Sun Microsystems, Inc.	Solaris™
• The Santa Cruz Operation, Inc. (SCO)	UnixWare
• Tandem Computers	Non-Stop Kernel
Network OS	
• Novell, Inc.	NetWare 4.0
Mainframe OS	
• International Business Machines	MVS

Figure 1. Companies licensing the Java Base Platform to embed in operating systems.

JavaChip™ Family

JavaSoft is working with Sun Microelectronics™ to develop the picoJava™, microJava™, and UltraJava™ family of microprocessors. The first of these, picoJava, is actually a standard specification for the design of a microprocessor that supports the Java Virtual Machine; this design is available for licensing to chip manufacturers. This design is a new architecture that is not SPARC-based—it is optimized for the unique demands of Java, such as multithreading and garbage collection.

The microJava and UltraJava are actual chips being developed by Sun Microelectronics based on the picoJava design. These chips have the Java Virtual Machine and Java Embedded API implemented in silicon, and vary in

[REDACTED]

[REDACTED]

dgolub
The Java Language is the the entry ramp to the Java Platform. Programs written in the Java Language and then compiled will run on the Java Platform.

[REDACTED]

[REDACTED]

dgolub
Java Application Programming
Interface (Java API)



[REDACTED]

[REDACTED]



dgolub
Combined, these parts provide
an end-user runtime environment for
deploying Internet and intranet
applications



[REDACTED]

dgolub
has a minimal set of
API required to run basic applets
and applications. This minimal set is
known as
the Java Applet API, or Java Base
API.

[Redacted]



dgolub
Developers who write to this
minimum
set can feel secure that the program
will run anywhere without the need
for
additional class libraries.

dgolub
The API targeted for this platform is
called the Java Embedded API.

[Redacted]

[Redacted]

[Redacted]

