The Java™ Virtual Machine Specification

Tim Lindholm
Frank Yellin

ADDISON-WESLEY

An imprint of Addison Wesley Longman, Inc.

Reading, Massachusetts • Harlow, England • Menlo Park, California
Berkeley, California • Don Mills, Ontario • Sydney
Bonn • Amsterdam • Tokyo • Mexico City

Duke™ designed by Joe Palrang.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the United States Government is subject to the restrictions set forth in DFARS 252.227-7013 (c)(1)(ii) and FAR 52.227-19.

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

Sun Microsystems, Inc. (SUN) hereby grants to you a fully-paid, nonexclusive, nontransferable, perpetual, worldwide limited license (without the right to sublicense) under SUN's intellectual property rights that are essential to practice this specification. This license allows and is limited to the creation and distribution of clean room implementations of this specification that: (i) include a complete implementation of the current version of this specification without subsetting or supersetting; (ii) implement all the interfaces and functionality of the standard `java.*` packages as defined by SUN, without subsetting or supersetting; (iii) do not add any additional packages, classes or methods to the `java.*` packages; (iv) pass all test suites relating to the most recent published version of this specification that are available from SUN six (6) months prior to any beta release of the clean room implementation or upgrade thereto; (v) do not derive from SUN source code or binary materials; and (vi) do not include any SUN binary materials without an appropriate and separate license from SUN.

Sun, Sun Microsystems, Sun Microsystems Computer Corporation, the Sun logo, the Sun Microsystems Computer Corporation logo, Java, JavaSoft, JavaScript, and HotJava are trademarks or registered trademarks of Sun Microsystems, Inc. UNIX® is a registered trademark in the United States and other countries, exclusively licensed through X/Open Company, Ltd. All other product names mentioned herein are the trademarks of their respective owners.

THIS PUBLICATION 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 PUBLICATION 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 PUBLICATION. SUN MICROSYSTEMS, INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS PUBLICATION AT ANY TIME.

Text printed on recycled and acid-free paper

ISBN 0-201-63452-X
1 2 3 4 5 6 7 8 9-MA-00999897
First printing, September 1996

CHAPTER 1

Introduction

A Bit of History

JAVA is a general-purpose concurrent object-oriented programming language. Its syntax is similar to C and C++, but it omits many of the features that make C and C++ complex, confusing, and unsafe. Java was initially developed to address the problems of building software for networked consumer devices. It was designed to support multiple host architectures and to allow secure delivery of software components. To meet these requirements, compiled Java code had to survive transport across networks, operate on any client, and assure the client that it was safe to run.

The popularization of the World Wide Web made these attributes of Java much more interesting. The Internet demonstrated how media-rich content could be made accessible in simple ways. Web browsers such as Mosaic enabled millions of people to roam the Net and made Web surfing part of popular culture. At last there was a medium where what you saw and heard was essentially the same whether you were using a Mac, PC, or UNIX machine, and whether you were connected to a high-speed network or a slow modem.

Web enthusiasts soon discovered that the content supported by the Web's HTML document format was too limited. HTML extensions, such as forms, only highlighted those limitations, while making it clear that no browser could include all the features users wanted. Extensibility was the answer.

Sun's HotJava browser showcases Java's interesting properties by making it possible to embed Java programs inside HTML pages. These programs, known as *applets*, are transparently downloaded into the HotJava browser along with the HTML pages in which they appear. Before being accepted by the browser, applets are carefully checked to make sure they are safe. Like HTML pages, compiled Java programs are network- and platform-independent. Applets behave the same

way regardless of where they come from, or what kind of machine they are being loaded into and run on.

With Java as the extension language, a Web browser is no longer limited to a fixed set of capabilities. Programmers can write an applet once and it will run on any machine, anywhere. Visitors to Java-powered Web pages can use content found there with confidence that it will not damage their machine.

Java has demonstrated a new way to use the Internet to distribute software. This new paradigm goes beyond browsers. We think it is an innovation with the potential to change the course of computing.

The Java Virtual Machine

The Java Virtual Machine is the cornerstone of Sun's Java programming language. It is the component of the Java technology responsible for Java's cross-platform delivery, the small size of its compiled code, and Java's ability to protect users from malicious programs.

The Java Virtual Machine is an abstract computing machine. Like a real computing machine, it has an instruction set and uses various memory areas. It is reasonably common to implement a programming language using a virtual machine; the best-known virtual machine may be the P-Code machine of UCSD Pascal.

The first prototype implementation of the Java Virtual Machine, done at Sun Microsystems, Inc., emulated its instruction set in software on a handheld device that resembled a contemporary Personal Digital Assistant (PDA). Sun's current Java release, the Java Developer's Kit (JDK) version 1.0.2, emulates the Java Virtual Machine on Win32, MacOS, and Solaris platforms. However, the Java Virtual machine does not assume any particular implementation technology or host platform. It is not inherently interpreted, and it may just as well be implemented by compiling its instruction set to that of a real CPU, as for a conventional programming language. It may also be implemented in microcode, or directly in silicon.

The Java Virtual Machine knows nothing of the Java programming language, only of a particular file format, the `class` file format. A `class` file contains Java Virtual Machine instructions (or *bytecodes*) and a symbol table, as well as other ancillary information.

For the sake of security, the Java Virtual Machine imposes strong format and structural constraints on the code in a `class` file. However, any language with functionality that can be expressed in terms of a valid `class` file can be hosted by the Java Virtual Machine. Attracted by a generally available, machine-independent platform, implementors of other languages are turning to the Java Virtual Machine as a delivery vehicle for their languages. In the future, we will consider bounded extensions to the Java Virtual Machine to provide better support for other languages.

Summary of Chapters

The rest of this book is structured as follows:

- Chapter 2 gives an overview of Java concepts and terminology necessary for the rest of the book.
- Chapter 3 gives an overview of the Java Virtual Machine.
- Chapter 4 defines the `class` file format, a platform- and implementation-independent file format for compiled Java code.
- Chapter 5 describes runtime management of the constant pool.
- Chapter 6 describes the instruction set of the Java Virtual Machine, presenting the instructions in alphabetical order of opcode mnemonics.
- Chapter 7 gives examples of compiling Java code into the instruction set of the Java Virtual Machine.
- Chapter 8 describes Java Virtual Machine threads and their interaction with memory.
- Chapter 9 describes an optimization used by Sun's implementation of the Java Virtual Machine. While not strictly part of the specification, it is a useful technique in itself, as well as an example of the sort of implementation technique that may be employed within a Java Virtual Machine implementation.
- Chapter 10 gives a table of Java Virtual Machine opcode mnemonics indexed by opcode value.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.