



An Overview of Java for the AS/400 System

Jennifer Bigus, IBM Corp.

Abstract

This white paper discusses what Java is and why it is a key piece of the application development strategy for the AS/400 system.

Because the AS/400 system is based on the concepts of object-orientation and platform independence, it is a natural fit for Java. The AS/400 Toolbox for Java and the AS/400 Developer Kit for Java will enable businesses to take full advantage of Java when developing AS/400 applications.

The AS/400 Toolbox for Java enables Java clients to access existing programs, commands, and data on the AS/400 system, thereby providing an easy entry into the Java world. The AS/400 Developer Kit for Java will make Java available for application deployment on the server. It is being developed with a focus on scalability so that it can support large-scale business object applications.

Introduction

Java is a hot, new programming language designed for today's networked world. This white paper discusses what Java is and why it is a key piece of the application development strategy for the AS/400 system. Read on to find out about Java on the AS/400 today and where it's going in the future.

Java Overview

Java is an object-oriented programming language developed by Sun Microsystems. It was originally designed for programming consumer electronics. The language was designed to be "architecture-neutral" so that it could run on the different computer chips used in the various consumer electronic devices. But Sun soon realized that the language had potential to do much more.

The architecture-neutral aspect of Java makes it ideal for programming on the Internet. It allows a user to receive software from a remote system and execute it on a local system, regardless of the underlying hardware or operating system. This is possible because of the interpreted nature of the language and the Java Virtual Machine.

Traditionally, the source code of a program is written in the programming language of choice and compiled into the machine code understood by a particular set of computer hardware. The Java compiler, however, does not generate machine code. Instead, it generates intermediate code called Java bytecodes. These bytecodes are interpreted by the Java interpreter, which executes the instructions on the particular hardware platform. The Java interpreter and run-time system are collectively called the Java Virtual Machine or JVM.

The bytecodes are what make Java programs portable. A program written in Java is compiled into bytecodes. These bytecodes can be transported across a network and executed on any system that implements a Java Virtual Machine.

An applet is a Java program designed to be included in an HTML Web document. The HTML document contains tags that specify the name of the Java applet and its Uniform Resource Locator (URL). The URL is the location at which the applet bytecodes reside on the Internet. When an HTML document containing a Java applet tag is displayed, a Java-enabled Web browser downloads the Java bytecodes from the Internet and uses the JVM to execute the code from within the Web document. These Java applets are what enable Web pages to contain animated graphics or interactive content.

Because Java applets can be downloaded from any system, security mechanisms exist within the JVM to protect against malicious applets. The Java runtime system verifies the bytecodes as they are downloaded from the network to ensure they are valid bytecodes and that the code does not violate any of the restrictions placed on Java applets by the JVM. Java applets are restricted in what kind of operations they can perform, how they access memory, and how they use the JVM. The purpose of the restrictions is to prevent a Java applet from gaining access to underlying operating system or data on the system.

But Java can be used for more than programs running within a browser. Java is a full-function programming language that can be used to write stand-alone applications that run outside of a Web browser.

One of the benefits of using Java for application development is that it is an object-oriented language. This means that the focus is on the data and the methods that operate on the data. The data and the methods comprise a class that defines the state and behavior of an object. With the exception of a few primitive data types like integers or floating point numbers, everything in Java is an object. Java also supports inheritance, which allows a new class to be created by extending an existing class. The new class can inherit the state variables and methods of the existing class and can add new variables and methods for new state and behavior.

In addition to the programming language constructs necessary for object-oriented program development, Java includes a rich set of predefined classes that are grouped together in packages. The early release of the Java Development Kit (JDK 1.0) contained the following packages:

- **java.lang**, which contains the base classes
- **java.io**, which contains classes for input and output to files and streams
- **java.net**, which contains classes for sockets programming and other networking
- **java.applet**, which contains the classes needed for applet programming
- **java.awt**, which contains the classes needed for GUI development

The current version of Java (JDK 1.1.4) includes these additional packages:

- **java.text**, which contains classes for internationalization, allowing applets and applications to be localized to different national languages and conventions
- **java.security**, which contains additional security features like digital signatures that allow an applet to be signed by originator and cryptography so that information can be encrypted before traveling over the network
- **java.rmi**, which allows objects to be distributed across the network and called using remote method invocations
- **java.beans**, which is a cross-platform component model used to build applications using pluggable pieces
- **java.sql**, which allows database access using a standard SQL interface called JDBC (Java Database Connectivity)

Applications written using the JDK are portable. "Write once, run anywhere" has become the rallying cry of Java application programmers. Java applications developed on one system can be deployed on a different system without having to change or recompile the code. And Java is becoming ubiquitous, with Java Virtual Machines available or planned for every major hardware platform and operating system.

One of the downsides of the portable, interpreted nature of Java is performance. While the performance of interpreted Java code is better than scripting languages and is fast enough for interactive applications, it is slower than traditional languages whose source code is compiled directly into the machine code for a particular machine. To improve performance, Just-In-Time compilers (JITs) have been developed on many systems. A JIT compiler runs concurrently with the JVM and determines which methods within the Java code are called most often. These methods are compiled into machine code on-the-fly so that they do not need to be interpreted each time they are encountered within a program. Static compilers are also being developed that compile the Java source code into machine code that is executed without interpretation. This compilation is not done on-the-fly; it is a separate step in the program development process. And it is important to note that the machine code generated is not portable and will not execute on other platforms.

Why Java for the AS/400?

There is a great deal of synergy between Java and the architecture of the AS/400 system. Java is an object-oriented programming language. The AS/400 system and its predecessors have had an object-based architecture from the very beginning. The AS/400 knows objects!

The Java Virtual Machine and the platform independence it provides are also well-known concepts in the AS/400 world. The AS/400 system is the only computer system today that allows applications to move to a 64-bit architecture without any porting or migration effort. The AS/400 system's technology-independent machine interface (TIMI) has shielded AS/400 customers from hardware technology changes for nearly 20 years.

Until now, the majority of the focus has been on client applets written in Java. But Java is a full-function programming language that can be used to write server applications. Java gives AS/400 developers the opportunity to move to object-oriented programming and modernize their applications without the complexity inherent in other

object-oriented languages like C++. Java is, by design, simpler than C or C++. For example, Java does not allow the programmer to manipulate pointers. A Java program has a reference to an object, which may be a pointer to that object, but that reference cannot be used to step through memory by manipulating the pointer. Java also implements automatic garbage collection, which eliminates the memory leaks that can cause "out of memory" errors in C and C++ code today. Other complex features of C++ that make the language both difficult to learn and to debug do not exist in Java. These include multiple inheritance, templates, and operator overloading. This makes Java a much simpler language, shortening both the learning curve and the development cycle when compared to other object-oriented languages.

For the AS/400 customer who does not wish to write programs in Java, the bytecode portability ensures that "100% Pure Java" code will run without recompilation on the AS/400. This means that any application development environment can be used to develop the code and create the Java bytecodes. The application can then be deployed on any platform that includes a JVM that conforms to the Sun specifications.

For large IT departments, this means that a "one-size-fits-all" approach can be used for application deployment. Only one version of a Java program is needed, regardless of how many different computer platforms are supported within the organization.

Java is an open, cross-platform, de facto industry standard language. It solves Web programming problems that are difficult to solve with more traditional languages. It is the ideal language for network computing (thin clients) and the World Wide Web. The AS/400 system is on its way to becoming a premiere server for network computing. Java is an important piece of the overall AS/400 Internet/intranet strategy.

Java on the AS/400 Today

Technology previews of Java on the AS/400 will be available soon in the "Developer Tools" section of this Web site. These previews are ports of Sun's JDK 1.0.2 and JDK 1.1 to the AS/400 and are not intended for application deployment. The previews contain early versions of the JDKs on the AS/400 and do not contain all the standard Java packages and classes.

Because a technology preview is not optimized for the AS/400 and does not include any JIT or other compiler technology, the performance characteristics are similar to the initial Java offering from Sun when they first released the technology. For many applications, the performance of a technology preview will be slower than that seen on comparable platforms using JIT or other optimizing technology. The technology previews allow new Java function to be available on the AS/400 system as soon as possible, providing a "sneak preview" for AS/400 application developers.

Java applets and applications that access AS/400 programs and data from client workstations can be written today using the AS/400 Toolbox for Java. Java classes on the client can be used to access existing AS/400 applications and data, providing easy entry into Java development. A socket is used to connect to AS/400 servers, providing access to AS/400 resources such as the following:

- Remote commands
- Distributed program calls
- Data queues
- Integrated File System data

- Print
- Record-level access
- Database access via JDBC on the client

The classes are written entirely in Java and can be run on any platform that supports JDK 1.1 or later.

Strategy and Direction for Java on the AS/400

Java is a key application development language for the AS/400 system. As the Java technology evolves from Sun, the AS/400 system will take advantage of the new functions and features of the language.

There is an exciting future for Java on the AS/400. This section describes the current plans for Java technology; however, it is not a commitment to the function or performance of future products on the AS/400.

Currently, the AS/400 Developer Kit for Java is being constructed to support the JDK 1.1.4 version of Java. A Java Virtual Machine that resides below the TIMI is planned to enable fast interpretation and execution of Java code on the AS/400. In addition, a type of static compiler is being developed that will generate RISC machine code from Java bytecodes. The Java transformer will enable direct execution of Java on the AS/400 system without the overhead of interpretation.

High-performance garbage collection is also being developed for the AS/400 system to improve both the performance and the scalability of Java. This advanced garbage collection algorithm allows Java to scale to the large numbers of objects expected when running business applications on the server. Over time, Java will become more integrated with and tuned for OS/400 to meet the requirements of performance and scalability on the server without compromising the cross-platform portability of the language.

Other technology is being developed that will allow GUI applications to run on the AS/400 system without modification. Support is planned for Java on the AS/400 that intercepts GUI requests coming from a Java program and reroutes the requests to an attached workstation running its own JVM. The workstation will then interpret and display the java.awt graphical components. This will allow server programs that have graphical interfaces for configuration or tuning to run on the AS/400 without modification.

OS/400's unique single-level-store architecture is also being exploited to give Java objects on the AS/400 an advantage not available on any other platform. Java objects on the AS/400 system can be full-fledged system objects, which will allow them to be persistent, shared, secure, backed up, and restored. Therefore, the AS/400 system will be able to offer persistent Java objects with performance and support that is unparalleled in the industry. The AS/400 single-level-store technology permits Java objects to be stored in their object form without the performance and maintenance overhead of two-level-store operating systems.

Because of its support for objects and its built-in virtual machine, the AS/400 system is positioned to take full advantage of Java as it evolves from its current Web-focused usage to a full commercial application environment. The strengths of the AS/400 will be combined with Java's object-oriented, network computing technology to provide

business solutions.

Summary

Java is the language of choice for programming in today's network computing environment. It allows true portability of applications between platforms without modification or recompilation. It is an open, cross-platform industry standard that is supported by all the major players in the computer industry today.

Because the AS/400 system is based on the concepts of object-orientation and platform independence, it is a natural fit for Java. The AS/400 Toolbox for Java and the AS/400 Developer Kit for Java will enable businesses to take full advantage of Java when developing AS/400 applications.

The AS/400 Toolbox for Java enables Java clients to access existing programs, commands, and data on the AS/400 system, thereby providing an easy entry into the Java world. The AS/400 Developer Kit for Java will make Java available for application deployment on the server. It is being developed with a focus on scalability so that it can support large-scale business object applications.

[Technology previews](#) of the AS/400 system will be available soon in the "Developer Tools" section of this Web site. Beta code for the AS/400 system is available at www.as400.ibm.com/ (take the Software and Java links).

Java is a trademark of Sun Microsystems, Inc.

Other companies, products, and service names may be trademarks or service marks of others.

[Copyright](#) [Trademark](#)

[Java Feature Page](#) [Java Home](#)

[▶ IBM HOME](#) [▶ ORDER](#) [▶ EMPLOYMENT](#) [▶ CONTACT IBM](#) [▶ LEGAL](#)