Java in the Palm of Your Hand!

By John Papproth

WHAT IS JAVA?

The Java programming language celebrated its 10th birthday last year! The core Java language was developed by James Gosling at Sun Microsystems in 1995. Java contains just 53 words in its grammar. (Compare that to 135 in Visual Basic!) See FIGURE 1. Of course those 53 words are just the basis for hundreds of packages that support the Java Developer (see http://java.sun.com/j2se/1.5.0/docs/api). Java is one of the Object Oriented Programming (OOP) languages allowing for encapsulation (the class is the unit of development), single inheritance (one class can extend another), and polymorphism (methods are qualified by name and signature, objects can be addressed by type). It is also one of the "curly brace" languages like C++. For programmers new to OOP, Java's use of single inheritance (one class - one parent) over multiple inheritance, reference types over pointers and automatic garbage collection of dynamic areas makes it an easier language to learn than C++.

WRITE ONCE, RUN ANYWHERE!

Write once and run anywhere! This is the basic philosophy of Sun Microsystems Java. A Java application is compiled twice. You execute the first compile during the development of your program to transform your source into an intermediate language called bytecode. The second compile is handled by the Java Virtual Machine (JVM) when your bytecode begins execution. The second compile is directed by the Just in Time or JIT compiler within the JVM. It is this process that makes Java portable between host systems. You can run anywhere as long as you can find a Java Virtual Machine (JVM) for your target machine that is compatible with your compiled bytecode. Luckily JVMs are available on a variety of platforms ranging from IBM's Mainframe Z/OS to your cell phone's operating system. The focus of this article is the development of small device applications using Java.

WHAT IS THE JAVA ENVIRONMENT?

In order to develop an application using any of the Java specifications (J2SE, J2EE, or J2ME) you will need to meet the following system requirements:

- Microsoft Windows XP or (unsupported) Linux-x86
- Java™ 2 SDK, Standard Edition (J2SE SDK), version 1.4.2 (or above) if you plan to do actual development
- Java™ 2, Standard Edition Runtime Environment (JRE), version 1.4.2 (or above) if you only plan to run the demonstration applications.

FIGURE 1: Java keywords and reserved words

abstract
assert **
boolean
break
byte
case
catch
char
class
const *
continue
default
do
double
else
enum ***
extends
false ****
final
finally
float
for
goto *
if
implements
import
instanceof
int
interface
long
native
new
null ****
package
private
protected
public
return
short
static
strictfp **
super
switch
synchronized
this
throw
throws
transient
true ****
try
void
volatile
while

* reserved but not currently used
** added in version 1.2
** added in version 1.4
*** added in version 5.0
**** reserved word

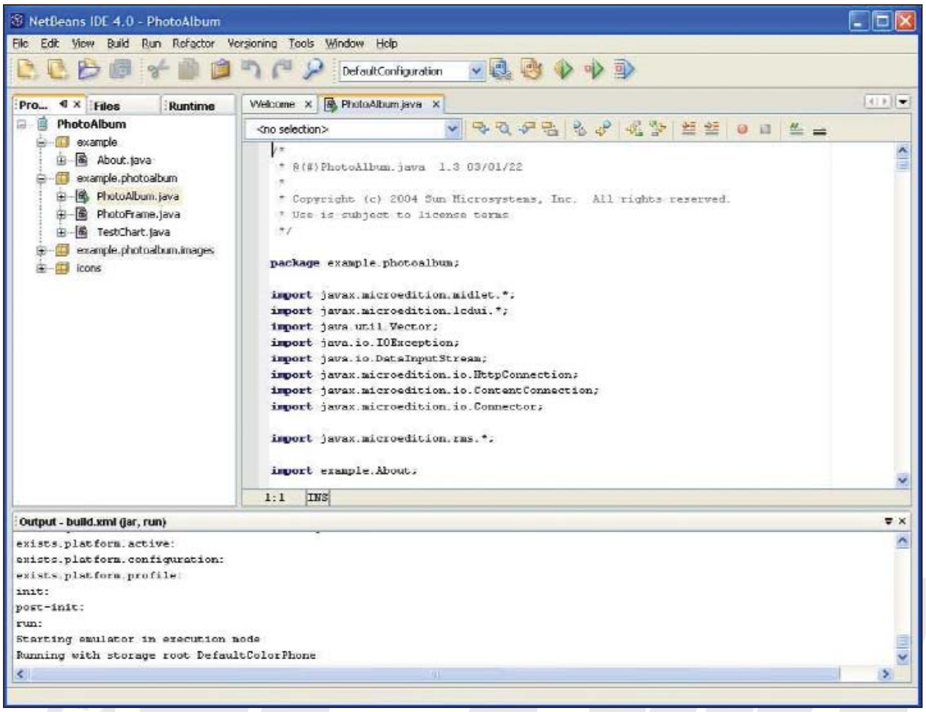
To download the SDK or JRE you want, go to http://java.sun.com/j2se/downloads.html

WHAT IS J2ME?

The Java 2 Platform, Micro Edition technology consists of a virtual machine and a set of APIs suitable for providing runtime environments for consumer devices (things you carry with you) and embedded electronics (things you plug into a wall). J2ME has two primary kinds of components: configurations and profiles.

Configurations are composed of two low-level APIs and optimized virtual machines targeted at two broad categories of devices based on the memory available. Those devices with 128-512K of memory available for the Java technology environment (and applications) use the CLDC - Connected Limited Device Configuration (CLDC) defined by Java Specification Requests (JSR): JSR 30, JSR 139. Java Specification Requests (JSRs) are the actual descriptions of proposed and final specifications for the Java platform (see http://jcp.org/en/jsr/overview). Devices with 512K+ available for the Java technology environment (and applications)

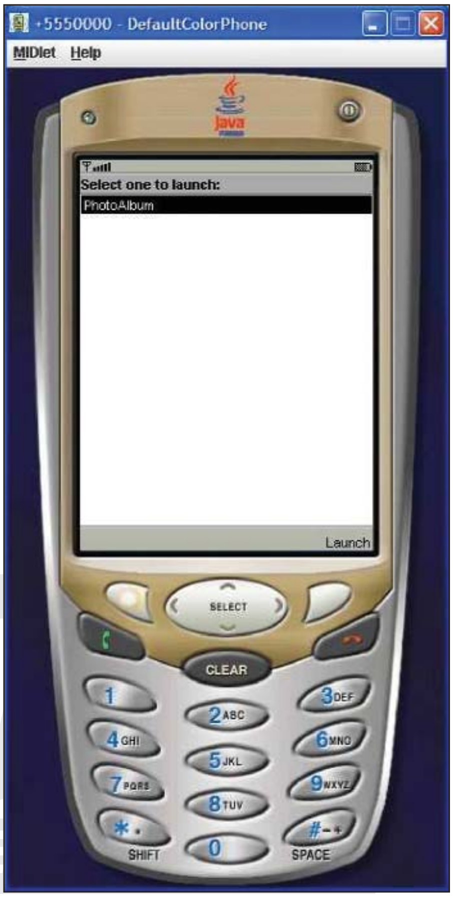
FIGURE 2: NET BEANS IDE



```
/*
 * @(#)PhotoAlbum.java  1.3 03/01/22
 *
 * Copyright (c) 2004 Sun Microsystems, Inc.  All rights reserved.
 * Use is subject to license terms
 */

package example.photoalbum;

import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import java.util.Vector;
import java.io.IOException;
import java.io.DataInputStream;
import javax.microedition.io.HttpConnection;
import javax.microedition.io.ContentConnection;
import javax.microedition.io.Connector;

import javax.microedition.rms.*;

import example.About;
```

```
exists.platform.active:
exists.platform.configuration:
exists.platform.profile:
init:
post-init:
run:
Starting emulator in execution mode
Running with storage root DefaultColorPhone
```

FIGURE 3: HANDHELD EMULATOR



use the CDC - Connected Device Configuration (CDC) defined by JSR 36, JSR 218.

Profiles are built upon a configuration. They provide a collection of APIs necessary to provide a complete runtime environment for a specific kind of device. An example of a profile is the MIDP - Mobile Information Device Profile (MIDP) defined by JSR 37, JSR 118.

In addition to configurations and profiles, the J2ME also contains groups of optional packages, such as API specifications that detail a set of classes and methods targeted at a certain kind of technology (media, for example).

Configurations, Profiles, Optional Packages…hey I just want to code something!

Ok, if you'd rather just get started, a good FREE place to begin is by downloading and installing the Java 2, Micro Edition (J2ME) Wireless Toolkit 2.2 Release. Remember that the J2SE SDK must already be installed (see "What is the Java Environment" earlier in this article).

The minimum hardware requirements for installing the Toolkit are:

- 50 MB hard disk space
- 128 MB system RAM
- 800 MHz Pentium III CPU

MIDP FOR THE PALM OS

Palm also provides Java support for small Palm O/S devices (like the Zire 32) through the MIDP for Palm OS. According to http://java.sun.com/products/midp4palm it lets Palm OS users run the same Java technology-based applications ("Java applications") that are available for other MIDP-compliant devices, such as mobile phones and pagers. Because it follows Palm OS user-interface conventions, MIDP for Palm OS creates a user experience that minimizes training requirements.

The Palm/OS system uses PRC (Palm Resource CODE) files as the installation unit. You can think of these as .EXE files on a desktop or as load modules on a mainframe. Before you can run any of the sample applications you will need to "hotsync" the MIDP.prc file to your Palm device. This will provide the necessary Java VM for the Palm OS.

PICK AN IDE, OR NOT

IBM's WebSphere Studio Device Developer provides an integrated development environment (IDE) for building, testing, and deploying Java™ 2 Micro Edition (J2ME™) applications that run on wireless devices such as cellular telephones, personal digital assistants (PDA), and handheld computers. The developer product is not free but the runtime environment is very inexpensive.

NetBeans IDE (see FIGURE 2) and Mobility Pack is an easy place to start. It is free and contains device emulators (see FIGURE 3).

An IDE is not required to do development. All of the Java utilities that are "front-ended" by the IDE are also available as command line utilities. If you have an ASCII editor, like Windows NOTEPAD, and a command prompt you are ready to write your first application.

THINGS TO CONSIDER (SIZE, SPEED, AND SECURITY)

The screen landscape is smaller in a mobile application. You will need to reduce any clutter in your interface design to make up for the relative difference in screen landscape between a desktop application and a low end handheld device. You can do this by providing a guide for your user through the use of prompts and menus.

Mobile applications have high latency. They can be slow. They can be interrupted easily. Provide progress bars so that your user is aware of the activity and to discourage them from "bailing out" too soon. Design your application to allow the user to cancel activities without causing inconsistencies between the client and server. Access the network when it is needed. Do not attempt to make a connection at the start of your application and drop it when the application completes. Provide client side validation to reduce long meaningless trips to the server!

Your application has left the building! You are designing applications that will be walking around with the user and with your data. Security becomes very important and any temporary client side storage of information on the handheld becomes critical data that you must protect or destroy when you are finished with it.

GLOSSARY

J2ME—Java 2 Micro Edition
CDLC—Connected Limited Device Configuration
MIDP—Mobile Information Device Profile
JAD—Java Application Descriptor used to install J2ME applications
JAR—Java Archive file
PRC—Palm Resource CODE (PRC) executable files

FOR MORE INFORMATION

The Java language is owned by Sun Microsystems, Inc. It makes sense to start looking for more information on Sun's Java Developers' Site.

http://java.sun.com/j2me/
http://java.sun.com

This article also discussed the Palm/OS system and it's relationship to Java. You can get additional information by going to the Palm site:

http://pluggedin.palmone.com/regac/pluggedin/Java.jsp

One of the platforms for enterprise development has been IBM's WebSphere. IBM has introduced "WebSphere Everyplace Micro Environment" to support the J2ME specification.

http://www-306.ibm.com/software/wireless/weme/

O'Reilly books have a section on the Java Language and on J2ME development that I have found useful.

http://www.oreilly.com/catalog/j2meanut/index.html
http://www.onjava.com/

In case you think of Java as "one of those" languages, think again! Java is also available on the mainframe under the OS/390 and ZOS systems from IBM.

http://www.redbooks.ibm.com/redbooks/SG245619.html

NaSPA member John Papproth is a both a MCSD.NET (Microsoft Certified Solution Developer) and SCJP (Sun Certified Java Programmer). He is a consultant for Bass & Associates Inc. a Business Solutions Provider in Omaha NE. John has taught courses in both the Microsoft .NET language suite and the Sun Microsystems Java Language.