Java ME Frequently asked Questions

General questions

(1) What is Java 2 Micro Edition?
To quote Sun's website:

A highly optimized Java runtime environment, JavaMETM technology specifically addresses the vast consumer space, which covers the range of extremely tiny commodities such as smart cards or a pager all the way up to the set-top box, an appliance almost as powerful as a computer.
Basically, JavaME is an umbrella term that refers to the collection of virtual machines, APIs and related elements that provide Java to handheld and embedded computing devices.

(2) What is a configuration?
A configuration defines a virtual machine and the minimum APIs that must be present. In defining the virtual machine, the configuration specifies the minimum amount of memory that must be present as well as the minimum processor type and speed.

(3)What is a profile?
A profile refers to the APIs for a specific domain. For example, the MIDP (Mobile Information Device Profile, description below) defines the APIs that are made available for pervasive computing devices such as mobile phones, wireless PDAs and similar devices.

(4) Can I use more than one profile at a time?
Yes, provided that the virtual machine and device supports the functionality for the additonal profile. For example, OEMs can provide their own profiles in addition to the MIDP to allow MIDlets to access device-specific features of their device, such as the information contained in the SIM card on a phone. However, if such a profile were provide by an OEM such as Nokia, this profile wouldn't be portable to other devices, such as Motorola.

Another point to bring up is that, in order to use an optional API, it must already be present on the targeted device. For example, the MobileMedia API defines APIs for accessing such elements as the digital camera on a device. Developers can use these APIs to write camera-aware applications. However, merely writing an application against these APIs and loading the application on a device will not give you access to the camera. The APIs for the optional profile must be supported by the virtual machine itself as well. If support is not built into the virtual machine, then those optional features will not be available.

(5) What devices are currently JavaME-enabled?
A complete list of devices can probably be found by searching the Internet. For a list of all Nokia devices that support JavaME one can use the search feature on Nokia Developer, in the **Devices** section.

(6)Where does PersonalJava fit into JavaME?
PersonalJava was a previous implementation of Java for pervasive devices, based on Java 1.1. It is unrelated to JavaME. However, the Personal profile will fill the vacancy left by PersonalJava.

(7) I've ported the KVM/CVM to [some platform]. Can I distribute/sell my port?

When you downloaded the reference implementation, you agreed to the Sun Community Source License, which forbids you from distributing the source code for the reference implementation to anybody not also a licensee (section d, part i). You can only distribute binaries for evaluation purposes and cannot sell the ported VM (section d, part ii).

(8) What are the minimum fields that must be present in a MIDlet suite's manifest?

The manifest must contain at least the following fields:

MIDlet-Name*
MIDlet-Version*
MIDlet-Vendor*
MIDlet-# (where # is the MIDlet number, one for each MIDlet)
MicroEdition-Profile
MicroEdition-Configuration
* Value must exactly match those present in the JAD file.

(9) What are the minimum fields that must be present in a JAD file?
A Java Application Descriptor must contain, at the minimum, the following fields:

MIDlet-Name*

(C) Copyright Nokia 2013. All rights reserved.

MIDlet-Version*
MIDlet-Vendor*
MIDlet-Jar-URL
MIDlet-Jar-Size
* Value must exactly match those present in the manifest.

(10) I've heard the term J2ME, but not JavaME. What's JavaME?
At JavaOne 2005, Sun announced a new naming convention for Java technologies. With this change, the micro-edition technologies were renamed to be Java ME, dropping the "2" in the name to avoid confusing customers or building an expectation for some future J3ME.

Developer Community

Are there any JavaME-specific newsgroups, discussion forums or mailing lists?

There are several, infact.

NAME URL
comp.lang.java.programmer comp.lang.java.programmer
CLDC-FEEDBACK http://archives.java.sun.com/cldc-feedback.html
KVM-INTEREST http://archives.java.sun.com/kvm-interest.html
J2ME-CDC-INTEREST http://archives.java.sun.com/j2me-cdc-interest.html
Nokia Developer http://www.developer.nokia.com/
Howard Forums http://www.howardforums.com/

Where can I find archives for [some] mailing list on JavaME-related topics?
The following is an incomplete list of archives where past messages are stored for some JavaME-related mailing lists:

KVM-INTEREST mailing list archives
CLDC-FEEDBACK mailing list archives
J2ME-CDC-INTEREST mailing list archives
iDEN Developer's SDKLIST

The Connected Limited Device Configuration

What are the minimum requirements for the CLDC?
For a device to support the CLDC, it must have at minimum:

16-/32-bit RISC/CISC processor
160k of memory (128k of which is for the VM and APIs)
some form of network connectivity

I want/need to use native methods. How can I do this?
The simple answer is: you can't. JNI is not available within the resource constrained KVM.

Some people will then bring up the KNI as an alternative/solution. This is, however, an incorrect answer and not a solution. The KNI is a Sun-specific porting layer, introduced into the reference implementation for the MIDP 2.0 and the CLDC 1.0.4. The purpose of this layer is to make porting these and future reference implementations to new platforms, and cannot be used as a means of linking native code at runtime.

Does the CLDC support floating point operations?
The CLDC 1.0 specification does not support floating point math. However, support for floating point math was introduced with the CLDC 1.1 specification.

If you're working with a device that is built on the CLDC 1.0 specification and you need floating point math capabilities, you can download and use the MathFP library.

The Mobile Information Device Profile

What is the MIDP?
The MIDP defines a set of APIs for mobile devices, such as cell phones and low-end PDAs (higher-end PDAs will better benefit from more powerful profiles such as the Personal Profile).

What is the current version of the MIDP specification?
There are currently two versions of the MIDP specification: version 1.0 and version 2.0 (reference to as MIDP-NG or MIDP Next Generation). The 1.0 specification was the first profile released under the JavaME umbrella in 2000. The 2.0 specification was released in 2002 and addressed many of the shortcomings of the original specification. The 2.0 specification is also backwards compatible with 1.0.

What does the acronym LCDUI stand for?

(C) Copyright Nokia 2013. All rights reserved.

There are several sources of conflicting information on this subject. This link claims that it stands for "Limited Connected Device UI", while [illegible] that it stands for "Liquid Crystal Display UI", and JavaME Enterprise Development claims it stands for "Least Common Denominator UI".

Is MIDP available for the Palm OS?
In June, 2003, Palm announced an alliance with IBM to release IBM's WebSphere Micro Environment on their Tungsten line of handheld devices.

You can also download MIDP4Palm, which was the first MIDP platform released for the PalmOS.

Read this for a note on the previously announced PDA Profile.

I don't have Java installed in /usr/local/java/jdkx.xx/x86/bin! How do I build the MIDP reference implementation?
There are several macros used by the makefile to determine where elements exist on the system. One, ALT_BOOTDIR, tells the makefile where Java is installed. By redefining this macro in the commandline to point to where Java's installed on your system, the build should complete (barring any other differences between your system and the expections of the makefile).

Assuming your JDK is installed in /usr/lib/j2se/1.3 (the default for Debian with JDK 1.3 installed), you would use the following commandline:

make ALT_BOOTDIR=/usr/lib/j2se/1.3

Can I access the address book on my PDA from my MIDlet?
Not with MIDP. However, there are APIs available in the PDA Optional Packages for the JavaME Platform that will give you access to this information.

The Personal Digital Assistant Profile

When will the PDA profile be available?
The PDA Profile is now called the PDA Optional Packages for the JavaME Platform and consists of two separate, optional packages:

Personal Information Manager APIs - Enables the developer to access the native personal information management data that resides on the device, including the address book, calendar, and to-do lists.
FileConnection APIs - Gives the developer access to file systems on the device, such as external memory cards.
These APIs will be implemented by the VM vendor on other JavaME profiles, such as the MIDP and Personal (and Personal Basis) Profile, and will enhance the CLDC.

The Connected Device Configuration

What are the minimum requirements for the CDC?
For a device to support the CDC, it must have at minimum:

32-bit RISC/CISC processor
2M RAM
some form of network connectivity

The Foundation Profile

Why can't I find any UI classes in the Foundation profile?
The Foundation profile, when used by itself, is intended for embedded and headless devices; i.e., devices that have no user interface, such as routers and gateway devices.

There are other profiles, such as the Personal Profile, that are intended to run on top of the Foundation profile.

The Personal (and Personal Basis) Profile

What is the Personal Profile?
The Personal Profile is the JavaME replacement for PersonalJava.

What is the difference between the Personal Profile and the Personal Basis Profile?
The Personal Basis Profile (JSR-129)defines a minimal set of user interface APIs. The Personal Profile (JSR-62) builds on the APIs from the Personal Basis Profile.

The Personal Profile is intended to be the next step in the migration path from PersonalJava and, as such, needs to support the full feature set of PersonalJava. The Personal Basis Profile is intended to address devices less capable than those targeted by the Personal Profile, providing a scaled down API set.

(C) Copyright Nokia 2013. All rights reserved.

I'm not using RedHat Linux. How do I build the reference implementation?
The build environment requires a few macros be defined at the commandline to tell it the locations for your JDK, the shell to use and the location for specific GNU tools. The following table lists the macros and the values necessary to perform the build on a Debian Linux system:

MACRO NAME DESCRIPTION VALUE ON DEBIAN
JDK_HOME The root of your JDK installation. /usr/lib/j2se/1.3
SHELL The path to the shell used for executing commands. /bin/bash
CVM_GNU_TOOLS_PATH The location for the standard GNU toolset. /usr/bin

Device Communications

What is the Generic Connection Framework?
The Generic Connection Framework is the paradigm used for creating javax.microedition.io.Connection objects that support different protocols. A URI defining the protocol and remote end-point for the network connection is passed to the javax.microedition.io.Connector class via one of the open() methods and a javax.microedition.io.Connection-type object is returned, ready to connect to the end-point.

I want to create a TCP/UDP/other connection with a server. Can I do this with MIDP?
Yes, if the device supports such a connection. The only required protocol is HTTP, but the OEM can provide other protocols as well.

How do I create a server socket with the GCF?
You don't. With the the GCF, there is no support for server sockets. However, you can instead create a javax.microedition.io.StreamConnectionNotifier instance and invoke its acceptAndOpen() API. The following snippet demonstrates how to do this:

```
public void waitForConnection() { // change the port
number to be whatever is appropriate for your application
StreamConnectionNotifier server = (StreamConnectionNotifier
)Connector.open("serversocket://:10000"); StreamConnection
conn = server.acceptAndOpen(); // do something useful with
the new connection }
```

Can I dial a telephone number from my MIDlet?
The MIDP 1.0 specification does not provide an API for doing this. The OEM, though, is free to add any additional profiles they wish to the mix.

The MIDP 2.0 specification includes platformRequest(String) method in the javax.microedition.midlet.MIDlet class. The telephone number can be passed in the form tel:<number> as defined in RFC 2806.

Can I determine the phone number from my MIDlet?
The MIDP 1.0 specification does not include APIs for accessing the phone number or SIM information for the device on which it's running. OEM's are free to add additional APIs for accessing this information, though.

The MIDP 2.0 specification includes platformRequest(String) which OEMs can use to provide this information if they so choose.

When I'm using HTTP connections, I get the message 'java.io.IOException: exceeded the configured maximum number of connections'. Why?
The answer involves the java.io.InputStream and java.io.OutputStream objects related to the javax.microedition.io.HttpConnection object in use. If you're using either input or output streams, it is necessary to close them after you've finished with the connection. Otherwise, the underlying implementation is most likely keeping the javax.microedition.io.Connection object open. Simply call close() on all related streams and it should solve the problem.

Also, be sure you're not using persistent connections if you're going to multiple servers during an application's sessions.

Can I determine the IP address for my device with my MIDlet?
No. Not all MIDP devices use IP-based networking, and the MIDP expert group didn't include an API for determining the address for those devices that do.

How do I setup persistent HTTP connections with a server?
In HTTP 1.1, persistent connections are the default behavior required by the specification. So, no special coding is required to suggest to the device that the underlying connection should be maintained between HTTP requests.

See the next question for information on how to turn off persistent connections.

Then how do I tell the device I don't want to maintain persistent connections?
When setting up the connection object, set the "Connection" property to "close". This will tell the server to tear down the connection after the response has been set.

(C) Copyright Nokia 2013. All rights reserved.

```
HttpConnection conn = (HttpConnection
)connector.open("http://foo.com/index.jsp"); // set the
Connection property if(!persistent)
conn.setRequestProperty("Connection","close");
```

Then can I re-use the same HttpConnection object more than once?
No, you cannot use the same object more than once. The HttpConnection object represents exactly ONE round-trip connection with the server and, as such, cannot be used for transmitting data more than one.

The connection object can exist in one of three states: 1) Setup, 2) Connected and 3) Closed and, when it moves to a new state, it can not return to a previous state. So, when data is sent to the server, the connection has moved from the setup state into the connected state. In the connected state, the connection can only receive data from the server and cannot send any more data to the server. When the incoming data has been received and the InputStream objects closed (or the connection itself is closed, but I'm assuming you were diligent and closed your input and output streams first <g>), the connection moves into the closed state and no further data can be read from the connection. Any attempt to use methods from a previous state on a connection object will result in an IOException being thrown.

Can I send SMS/MMS messages from my MIDlet?
You can send SMS messages, provided the device hosting your MIDlet supports JSR-120, the Wireless Messaging API 1.0. If your device supports version 2.0 of this specification, then you will be able to send MMS messages as well.

User Interfaces

Can I create new form elements?
Not with MIDP 1.0. According to the specification, javax.microedition.lcdui.Form only has APIs for appending elements using an instance of java.lang.String (which is a shortcut to creating an instance of javax.microedition.lcdui.StringItem, javax.microedition.lcdui.Image (which is a shortcut to creating an instance of javax.microedition.lcdui.ImageItem or javax.microedition.lcdui.Item. There's no way to create a new subclass of javax.microedition.lcdui.Item since the single abstract method, paint(), has package (default) protection. It is also against the licensing agreement (per section e, subsection iv) to create a new subclass in the javax.* package hierarchy. Finally, even if the licensing agreement were ignored, most VMs will ignore any class in the java[x].* hierarchy.

In MIDP 2.0, the class javax.microedition.lcdui.CustomItem was introduced. Subclassing this class will allow you to create new elements that can be added to forms.

I can create an Image from an array of bytes. Can I do the reverse and get an array of bytes from an Image?
Unfortunately, no. There's no API available for extracting anything more than the image's width, height, an instance of javax.microedition.lcdui.Graphics and mutability.

With the MIDP 2.0 you can get the raw bytes from an instance of javax.microedition.lcdui.Image through the getRGB() API.

How do I get notification of high level key events from within a Form, TextArea or List within the MIDP?
The high level user interface APIs in javax.microedition.lcdui do not give you access to the low level key events. These events are only available to classes which extend javax.microedition.lcdui.Canvas.

There is an alternative, though it may not meet the requirements above. The javax.microedition.lcdui.ItemStateListener interface defines a contract for objects that will be notified of state changes within a javax.microedition.lcdui.Form object. However, there is no specific set of conditions under which these notifications are sent; i.e., it is up to the device/VM vendor to decide when and if notifications will be sent. So, notifications will vary greatly between different devices and VM implementations.

Can I highlight text in my TextField or TextBox object, or in a StringItem on a form?
No. There is no way to mark up the text being displayed by these components.

Can I assign Command objects to specific soft keys?
No, you can't.

When you create an instance of Command you can only specify a label, a command type and a priority. The MIDP implementation uses the type and priority of the commands to be displayed to determine how the commands will be presented to the user. It will also make a best effort to map keys to the appropriate button for that device; i.e., if the device normally displays OK buttons on the right side, then it will attempt to present your command of type Command.OK on the right side.

Also, not all MIDs have softkeys. For example, the RIM Blackberry displays the Commands as menu items. On such a device, there's no concept of a softkey. Instead, on such devices, the implementation will use the priority of the Commands as well as their type in order to group and sort them prior to displaying them to the user. So, be sure that each command within each group has a unique priority.

When I run my MIDlet in the emulator, my image loads fine. But, when I run on a device, the image isn't displayed.
Verify that your image file is present in your MIDlet suite's JAR file and that it's JAR entry is the same as your code expects; i.e., if you're using Image.createImage("/foo.png"), then there must be a JAR entry named /foo.png or else the MIDP implementation will throw an IOException. Remember,

(C) Copyright Nokia 2013. All rights reserved.

also, that filenames are case sensitive and that you have to use the full entry name for the image.

Security

Can I load a class from outside of my MIDlet suite?
No. The MIDlet is limited to only loading classes from within it's suite. This prevents hackers from loading dangerous classes. It also prevents a device from using precious bandwidth for loading classes over the air or via low-bandwidth network connections.

Runtime Activities

How do I read a file from my JAR?
To read a file from your MIDlet suite's JAR file, use the java.lang.class.getResourceAsStream() API.

How can I tell my device to automatically load my MIDlet when [something] happens?
With MIDP 1.0, there's no way to automatically launch a MIDlet either at device startup or upon reception of some data.

With MIDP 2.0, there's the concept of the push registry, which allows a MIDlet to register itself for network and timer initiated MIDlet activation. Take a look at the javax.microedition.io.PushRegistry class Javadocs for more details on using the push registry.

What happens to my threads when my MIDlet is suspended?
On most devices, the only thread that is suspended is the main application thread for the MIDlet. Any other thread that was started by the MIDlet will most likely (but not always) continue running.

What system properties are available to determine the device type at runtime?
The following system properties are available, as of MIDP 2.0, to help determine the type of device on which the MIDlet suite is running:

microedition.platform Identifies the device type or host platform, or returns null if the value is undefined
microedition.encoding The default character encoding.
microedition.configuration The configuration this device has implemented.
microedition.profiles The profiles this device has implemented.
microedition.locale The current locale for the device. This value may be null.

I would like to get the physical location of the handset. Is this possible?
To get the handset's location, you would have to have a handset that supports the Location API for JavaME. With this API you can get the location for the handset at runtime. Unless the OEM has implemented this API on the handset, you will not be able to determine the physical location of the device.

Developing Software For JavaME

What are the minimum tools I need to begin development?
Minimally, you will need:

JDK version 1.3 or higher
the reference implementation for the targeted profile
the reference implementation for the targeted configuration
the ANT build utility (click here for an example build.xml file)
The reference implementations provide the APIs for compiling the application. They also provide the preverifier, which performs the initial verification of the created byte code. This saves the device from having to perform initial verification, which can be resource expensive. It also inserts extra information into the byte code which is needed by the VM, but which is ignored by the J2SE VM, so compiled classes can be shared between JavaME and J2SE.

This all said, I would now recommend instead that you download Sun's Wireless Toolkit 2.x or later, Eclipse 3.x or later and EclipseME. This gives the most complete environment for JavaME development for the price.

What would a minimal ANT build.xml file look like?
I've removed the ANT example from the FAQ. It is recommended that, rather than using this example for most tasks, you use the Antenna suite of ANT tasks. With Antenna, all of the more awkward tasks, such as preverification, obfuscation and the launching of emulators, have already been taken care of for you and are far easier to integrate into your build environment.

I installed the Sun Wireless Toolkit. Why can't I run the sample MIDlets?
This answer assumes you're running in a Unix/Linux environment, and installed using the root account.

When the toolkit was installed, it created the directory structure with root as owner (assuming you didn't install the product as yourself in your home directory, or one to which you have write access). The file access permissions for the tree are root.[staff on my system] rwxr-sr-x. As you can tell from this, only members of the owning group can read the directories where the toolkit was installed. In addition, only root can write to the directory tree. So, unless you are running the toolkit as root (not a very good practice at all) you will not be able to run any of the samples for the toolkit until you make some

(C) Copyright Nokia 2013. All rights reserved.

changes to the directory structure.

The best thing to do is to go into the JavaMEwtk/apps directory and change the permissions to allow the owning group to write to the directories as well. From the installation directory, execute the command:

find apps -type d -exec chmod 775 {} \;
This will descend into each directory and change the permissions to allow members of the owning group to write to the directories, but will not allow anybody to change the files in the directories. This way, the source code will still be read-only, but users will be able to compile and execute them.

What would the MIDP equivalent of Hello World! look like?
import javax.microedition.lcdui.Command; import
javax.microedition.lcdui.CommandListener; import
javax.microedition.lcdui.Display; import
javax.microedition.lcdui.Displayable; import
javax.microedition.lcdui.Form; import
javax.microedition.midlet.MIDlet; public class HelloWorld
extends MIDlet implements CommandListener { public
HelloWorld() { } public void startApp() { Form form = new
Form("Hello world!"); form.addCommand(new
Command("Exit",Command.EXIT,1));
form.setCommandListener(this);
Display.getDisplay(this).setCurrent(form); } public void
pauseApp() { } public void destroyApp(boolean
unconditional) { } public void commandAction(Command
c,Displayable d) { if(c.getCommandType() == Command.EXIT) {
notifyDestroyed(); } } }

When I attempt to obfuscate my MIDlet suite, Retroguard dies. Why?
It is a problem with the way obfuscation works in Java. Most obfuscators will begin altering the bytecode by traversing the methods and classes used by your MIDlet, and then examines the methods and classes used by those methods, to obfuscate the code. When RetroGuard hits a native method (for example, one within the MIDP API that's implemented in the KVM rather than in Java code), it cannot handle the situation and returns an error.

If you're using an archive or subdirectory in your classpath for RetroGuard that contains a working version of the MIDP API, RetroGuard is hitting one or more of the above-mentioned native code references. It is this that is causing the failure during obfuscation.

The solution is to use the file emptyapi.zip, which comes with the Wireless Toolkit. This archive provides all of the public interfaces used by the MID Profile without any of the underlying implementation code or references to any native methods.

When I preverify my classes, I get the message "Class loading error: Bad major version number". Why?
My guess is that you're using JDK version 1.4 or higher. The problem is that the byte code format generated by the 1.4 and later compiler is incompatible with earlier versions of the preverify executable that you're using.

To fix the problem, upgrade your toolkit to the latest version and see if that fixes the problem. If you're using the latest tool, or if your tool vendor hasn't upgraded their tools, then you can use the following commandline:

javac {whatever arguments you normally use} -target 1.1
{your more arguments}
This will cause the compiler to produce pre-1.4 byte code.

If you're using J2SDK 5, then you'll want to add the following commandline argument instead:

javac {whatever arguments you normally use} -source 1.2
-target 1.2

How do I use JDWPI to debug a MIDlet running in the Wireless Toolkit's emulator?
To access the emulator using sockets, start the emulator and include the following command line argument:

-Xrunjdwp:transport=dt_socket,address=[the local
port],server=y,suspend=y
replacing [the local port] with an available port number. Then, give the same port number to your debugger in order to connect to the emulator.

To access the emulator using shared memory, start the emulator and, instead of specifying a port, specify the name of the shared memory page using the following command line argument:

-Xrunjdwp:transport=dt_shmem,address=[the shared memory
page],server=y,suspend=y
replacing [the shared memory page] with a unique memory pagename. Then give the same name to your debugger in order to access the shared memory.

How can I see debugging output from my Motorola iDEN phone?
You need to connect your iDEN phone to a serial port on your computer using a data cable. Then, from a terminal application (set to 19200bps, issue the following command:

 (C) Copyright Nokia 2013. All rights reserved.

AT+WS46=252;+WS45=0;+IAPPL=2;D
This will cause all output sent to System.out and System.err to be sent to the serial port and, finally, to your terminal application.

Writing games using JavaME

Is there an editor to help me build the tiled layers for my game backgrounds?
Yes, there are a few.

For Windows, there is Tile Studio, World Creator and Mappy.

Provisioning A Device

How do I download my MIDlet suite onto my cell phone or wireless device?
This depends on the device in question. There are two ways to put a MIDlet suite onto a device:

Over-the-air (OTA) provisioning
Provisioning via data cable
Over-the-air provisioning is when the application is downloaded from a web or portal site and installed onto the device. An overview of OTA provisioning can be found here. The basic flow is that the device retrieves the JAD file from a web or portal site and determines whether the MIDlet suite can install onto the device. If the suite is acceptable (i.e., it's new to the device, or it's the same or more recent version of an installed suite, and it's not too big [if there's a size restriction imposed by the carrier]) then the user is prompted if they want continue with the installation. If the user accepts, then the MIDlet suite is downloaded over the air and installed onto the device.

Provisioning via a data cable is more direct in its approach. Software is provided by the OEM and is used to connect to the device normally via a serial or USB cable connected to the device, such as the cradle for a Palm or RIM, or the data cable for the Motorola phones. Using this software, the developer browses the local file system and selects the JAR and JAD files to install onto the device. The software then negotiates a connection with the device and uploads the MIDlet suite.

What are the MIME types for JAR and JAD files?
In order for a web server to host a MIDlet suite, it has to be configured to send the property content types when sending a JAR and JAD file. To do so, the following MIME types must be configured for the server:

FILE TYPE EXTENSION MIME TYPE
JAR .jar application/java-archive
JAD .jad text/vnd.sun.j2me.app-descriptor

To set these values in Apache2, add the following lines to your /etc/apache2/apache2.conf file:

AddEncoding application/java-archive jar

AddEncoding text/vnd.sun.j2me.app-descriptor jad

I want to test my MIDlet suite on my PC. What can I use?
Well, the first answer is to use the emulators that ship with Sun's Wireless Toolkit, being sure to grab the appropriate version for the MIDP version you're targeting.

Additionally, nearly every device manufacturer has their own emulators available for their devices. Motorola, Nokia and RIM even ship either complete IDEs (such as Metrowerks Code Warrior and Nokia's Developer Suite for J2ME.

For those looking for a Java-based solution, there is an (incomplete at this time) emulator written in Java called ME4SE that is an emulator built of Java 2 Standard Edition. It includes the ability to run the MIDlet within an applet in a user's web browser. There is also the MicroEmulator project which has built a MIDP emulator using the JDK 1.1. This would allow MIDlets to be demonstrated from within older browsers.

Another tool that provides a pure Java environment for emulator a MID is the MPowerplayer SDK. To quote their website:

The mpowerplayer software developer kit includes the mpowerplayer as a standalone application. It's a pure Java emulator implementing MIDP 2.0 and MMAPI, suitable for integration with your favorite environment and IDE.

Motorola-Specific Questions

How do I determine the firmware version of my iDEN phone?
With your handset, press the following button sequence:

1. * [menu] [right navigation button]

How do I determine the firmware version of my V-series phone?
On my V400 (and my V300 at work) I press the following sequence:

 (C) Copyright Nokia 2013. All rights reserved.

- # 9 9 9 9 #
  Although this same information is also available if you go to MENU -> Settings -> Phone Status -> Other Information, where each field is selectable individually.

How do I handle incoming phone calls from Java?
On the i730 it's:

Settings -> Phone Calls -> Java App Control -> [your MIDlet]

Table of Contents
Nextel-Specific Questions

What do the letters NXTL stand for on the RIM Blackberry?
As your device connects to the network, you'll notice in the upper-right corner of the display the letters "NXTL". These letters will individually cycle through upper- and lower-case states before finally (hopefully) becoming all upper-case, which signals a good connection to the network.

But, why does it do this? What do these characters mean? I have the answer. After having spent some time on the phone with Nextel's technical support I was given this explanation for what each character stands for:

N Cellular network connectivity
X Walky-talky feature connectivity
T Packet data network connectivity
L Network registration

Nokia-Specific Questions

How do I install software onto my Nokia handset?
There are multiple ways to install sofware onto your Nokia handset:

Over-the-air provisioning
Via infrared connection or data cable
Via Bluetooth
To install over-the-air, you need to have a data plan with your wireless carrier. For Series 40 and 60 phones, go to the Services screen and select Bookmarks Setup a link to your JAD file (or to the page hosting the JAD file) by selection Options and then New bookmark. Once it's setup, go back to the Bookmarks screen and click the call button (or click Options and then Go to) to open the link. Then follow the prompts to download and install the MIDlet suite.

To install via irDA, you need to have installed the appropriate PC Suite software from Nokia's website. Once installed, you can connect the handset to your PC and use the Application Installer too to deploy your application onto your handset.

For Bluetooth installation, you need to first connect to your handset via Bluetooth. Then, simply push the JAR and JAD files onto the handset. You should receive a message stating that new files were received and you need only follow the prompts to complete the installation process.

How do I determine the firmware version of my Nokia phone?
With your handset, press the following key sequence:

1. * 0 0 0 0 *

Samsung-Specific Questions

On my Samsung A600 my MIDlet never gets notified when the user presses the END key.
The unverified answer (I've never talked with someone from Samsung) seems to be that, when the END key is pressed, the A600 does not notify the MIDlet but instead stops the VM itself from executing. So, rather than receiving a callback (either MIDlet.pauseApp() or Canvas.hideNotify()) notification, your application will just stop.

If you're writing a time-sensitive application, then you should probably put some sort of time checking into your code and attempt to detect such situations. In my case, I wrote a game that had the side-effect of the player losing on time when the END key was pressed. To solve the problem, I had my core loop grab the time when it started and the time when it was finished a single iteration. If that time value was greater than a normal update event, then it assumed the END key had been pressed and modified the game time accordingly.

SprintPCS-Specific Questions

I have a JAR/JAD that works fine in the emulator. But, when I download to my Sprint phone, I get a 90x error, and invalid descriptor message and the JAR/JAD won't download. Why?
The reason is going to be either of both of the following elements missing from your JAD file: either there's no MIDlet-Data-Size entry (which is necessary) or your MIDlet-Jar-URL does not have the full URL for the JAR file; i.e., it has filename.jar instead of http://yourwebsite/filename.jar. SprintPCS phones require the full URL for your MIDlet suite in order to download and install it to your handset.

(C) Copyright Nokia 2013. All rights reserved.

What are the sound APIs for SprintPCS's VM?

To play sounds on a Sprint device, you need to first create an instance of the com.sprintpcs.media.Clip class, providing it with either the full path to the sound file in your JAR, or an array of bytes that comprise the sound file.

Once you have a Clip, you then pass it to one of the com.sprintpcs.media.Player class's static methods:

play(Clip clip,int repeat), which plays the clip for the specified number of repetitions, but will end the clip should another one be played
playBackground(Clip clip,int repeat), which plays the clip for the specified number of repetitions. The sound will be paused when another sound is played, and will continue until that sound is completed.

T-Mobile-Specific Questions

Sometimes my T-Mobile handset has Internet connectivity and sometimes it doesn't. Why?
This answer assumes two things:

You're in an area where there is no T-Mobile coverage, and
You're in an area where Cingular and at least one other carrier is present (such as North Carolina)
If these assumptions are correct, then you need to manually select your network and choose Cingular. At least here in North Carolina, T-Mobile has no network coverage and so has a sharing agreement with Cingular. When I use a T-Mobile device, it sometimes selects AT&T rather than Cingular and will have no Internet connectivity. Simply switching it back to Cingular solves the problem, for me anyway.

Vodafone-Specific Questions

What does the MIDxlet-Resident JAD property represent?
This indicates whether the MIDlet will remain active in memory (resident) in the background and able to process events.

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.