

# Getting Started with MIDP Programming on Symbian OS

Author: Martin de Jode  
Date: August 2004  
Version: 0.3

## Introduction

This paper is intended as a starting point for developers planning to undertake MIDP development on Symbian OS. The aim of the paper is to smoothe the way for developers migrating to MIDP by pointing them in the direction of essential resources available from [Symbian Developer Network](#) and other portals.

## First steps: scoping your project

The first thing to remember is that MIDP is not J2SE or [PersonalJava](#) or Personal Profile (the J2ME successor to PersonalJava). [MIDP](#) was specified with highly constrained devices in mind so a lot of functionality is missing that you may have taken for granted in richer Java environments. Principle among these omissions is AWT/Swing and JNI. However, this doesn't mean you can't create rich applications using MIDP. MIDP supports its own GUI framework in the LCDUI, extended to add flexibility in [MIDP 2.0](#). Furthermore extra functionality is available to MIDlets via the J2ME optional packages of which Symbian offers a particularly rich (and ever expanding) [collection](#). Knowing what selection of optional packages your target platform supports is an important consideration. If your target doesn't implement a J2ME API to perform a task required by your application, then, with no JNI support available to MIDP application developers, there is no way for a MIDlet to break out of the Java sandbox and hook into native APIs.

So to summarise this section a few key points to bear in mind:

- Know the limitations of MIDP
- Know your target platform (which MIDP version and which optional APIs are supported)
- Design your project within those limitations OR opt for a different programming environment.

## Finding the Tools

So now assuming that MIDP fits the bill, its time to get down to some serious coding. The usual starting point in MIDP development is Sun's [J2ME Wireless Toolkit](#) (WTK). This easy-to-use, lightweight tool allows you to compile, preverify and run your MIDlet. It also provides various profiling options and other utilities for developing MIDP 1.0 and 2.0 applications. However, the WTK is not a full IDE (for instance it doesn't provide a debugger). As you get deeper into MIDP programming you'll want to use a fully fledged wireless enabled IDE. [Various J2ME MIDP enabled IDEs](#) are available from the major manufacturers.

Both the WTK and mobile IDEs facilitate the plug-in of 3<sup>rd</sup> party emulators (in addition to Sun's KVM which ships with the WTK). The advantage of using a 3<sup>rd</sup> party emulator is that it will more accurately model the implementation running on the target phone and provide implementations of any optional J2ME packages supported by the phone. [MIDP SDKs](#) for Symbian OS-based phones can be downloaded from Symbian Developer Network. Finally, don't forget to try application prototypes on the target hardware as early as possible in the development cycle. Emulators are never perfect and the earlier a glitch is detected the easier it is to trace and correct.

## Finding More Information

Being a Java standard there is a wealth of information available on J2ME from a large variety of sources. We'll briefly look at what's available from some of the main sources.

### **Sun**

Sun has a [portal](#) dedicated to J2ME development. Here you'll find a variety of tools, technical tips, white papers and example code.

### **Java Community Process**

To obtain a specification in javadoc format your first point of call is the [JCP portal](#) which hosts all Java Specification Requests (JSRs). A few key J2ME JSRs are listed below:

- CLDC 1.0      JSR 030
- CLDC 1.1      JSR 037
- MIDP 1.0      JSR 118
- MIDP 2.0      JSR 139

### **Symbian Developer Network**

Specifications are very useful for a detailed understanding of the capabilities of an API, however sometimes they can be short on example code. To help developers to get started programming an API the [Symbian Developer Network](#) provides introductory white papers covering the main APIs supported by Symbian's Java platform. These practical guides detail sample code all of which has been tested out on actual phones. Furthermore they indicate what functionality defined as optional under the relevant JSR specification is supported on the phone. Also they point out known defects and possible workarounds.

Symbian Developer Network also provides a [Knowledgebase](#) with over a thousand FAQs, covering useful programming hints, known issues and workarounds. The Knowledgebase can be filtered to just display Java FAQs. Symbian Developer Network also hosts free [discussion forums](#) covering a wide range of topics including Java.

### **Forum Nokia**

[Forum Nokia](#) contains a wealth of material covering all Nokia phones, including sections devoted to Symbian OS and Java including tools and SDKs, "How To" documents, "Known Issue" documents and discussion forums.

### **Sony Ericsson Developer World**

[Sony Ericsson Developer World](#) is the developer support forum for Sony Ericsson phones, including Symbian OS phones such as the P800 and P900. Sony Ericsson Developer World contains a wide range of J2ME material including Tips and Tricks, discussion forums, white papers and tools.

### **Books**

There are a wide range of books available on J2ME MIDP programming. However, check that your preferred title is a recent edition featuring the latest version of Mobile Information Device Profile, MIDP 2.0. Those developers serious about MIDP programming on Symbian OS will find "[Programming Java 2 Micro Edition on Symbian OS](#)" an indispensable addition to their library.

## Alternatives to MIDP

We started off this paper by reminding the reader that MIDP does impose limitations on developers. What are the alternatives if MIDP just doesn't provide the environment you need? One obvious option is to opt for native development with C++. This naturally gives you full access to all the functionality available in the Symbian platform. An alternative Java route to explore is the Personal Java/Profile. Although Symbian focused its Java implementation on CLDC MIDP from Symbian OS, Version 7.0s onwards, some phones based on earlier releases, particularly those based on UIQ reference design, provide a PersonalJava Application Environment. Furthermore recognising that some more fully featured devices, particularly those targeted at the enterprise, may need a more heavyweight Java solution Symbian has worked with partners and licensees to ensure a CDC/ Personal Profile implementation is available to licensees who need it. For instance the Symbian OS-based Series 80 Nokia 9500 provides IBM's J9 CDC/Personal Profile Java platform as well as Symbian's standard CLCD/MIDP based offering.

Currently the vast majority of Java enabled phones support MIDP, so choosing to target the Mobile Information Device Profile opens up your application to the widest possible market. However CDC/Personal Profile is gaining traction in the wireless space, particularly for high-end phones targeted more at the enterprise.

## Summary

Rather than attempt to provide an exhaustive "Getting Started" manual the aim of this paper is to steer the reader to the essential resources and information that a newcomer to programming MIDP will need. After some advice on scoping your MIDP project we then moved on to discuss where to find the essential tools for the job. The following sections then looked at where to find more detailed coding guides, API specs, books and support. Finally, we suggested alternative programming environments in the event that MIDP just doesn't provide what's needed.

## Resources

[Symbian Developer Network](#)

[Differences between PersonalJava and MIDP Java Environments](#), White Paper

[Mobile Information Device Profile Overview](#)

["What's New in MIDP 2.0"](#), White Paper

["What's new for Java Developers in Symbian OS v8.0"](#), White Paper

[Sun J2ME Wireless Toolkit](#)

["Java MIDP Development Tools"](#), White Paper

[MIDP SDKs for Symbian OS phones](#)

[Sun's J2ME Developer Portal](#)

[Java Community Process Website](#)

[Symbian Developer Network Knowledgebase](#)

[Symbian Developer Network Discussion Groups](#)

[Forum Nokia](#)

[Sony Ericsson Developer World](#)

[“Programming Java 2 Micro Edition on Symbian OS”, Wiley, 2004](#)

[Back to Developer Library](#)

Want to be kept informed of new articles being made available on the Symbian Developer Network?

[Subscribe to the Symbian Community Newsletter.](#)

The Symbian Community Newsletter brings you, every month, the latest news and resources for Symbian OS.