

✓ This tick symbolizes the featured content on the Forum Nokia Wiki. Featured articles are considered to be good quality articles in the Forum Nokia Wiki, as determined by the [Forum Nokia Wiki administrators](#). Before being listed here, these articles are reviewed for accuracy, neutrality, completeness, and style.

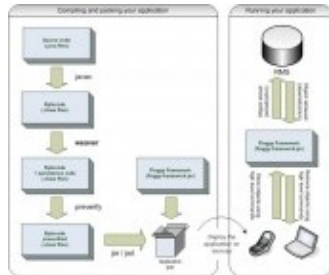
A small tick (✓) in the top-right corner of an article indicates that the article is a featured article.

View by category: [All](#) | [Symbian C++](#) | [Open C/C++](#) | [Java](#) | [Qt](#) | [Python](#) | [Flash Lite](#) | [WRT Widget](#) | [Maemo](#) | [Other](#)

Floggy Framework: RMS no more!

Week 47 - November 15th 2009

This week's featured article, [Floggy Framework: RMS no more!](#), is written by [Thiagobrunoms](#).



RMS (Record Management System) is the well-known API to provide data persistence in the [Java ME](#) platform. However, due to its limitations and some complexities to cope with Java I/O, some projects were proposed to facilitate the development of JavaME applications that need to store data.

Floggy is a free object persistence framework for J2ME/MIDP applications. The main goal of this framework is to abstract the data persistence details from the developer, reducing the development and maintenance effort.

The article illustrates the advantages of using Floggy framework over the conventional RMS. Further, it also explains how to use Floggy Framework in your Java ME applications.

[Read the article](#) and start using Floggy Framework in your Java ME applications.

Parsing XML files with NanoXML

Week 45 - November 1st 2009

This week's featured article, [Parsing XML files with NanoXML](#), is written by [Thiagobrunoms](#).



XML is a very flexible standard for describing tree-based data format. It is very useful for organising data in a structure that can then be read by other applications.

This article shows how to use NanoXML, an API designed for parsing XML data in [Java ME](#). The article comes with easy to understand example accompanied by screenshots and serves as a very useful guide for getting started with XML using mobile Java, demonstrating how to create and read XML files.

[Read the article](#) to get a quick introduction to XML parsing in Java ME both in general as well as using NanoXML.

[Microlog: A Log4j-based tool for the JavaME platform](#)

Week 42 - October 11th 2009

This week's featured article, [Microlog: A Log4j-based tool for the JavaME platform](#), is written by [Thiagobrunoms](#)



[Microlog](#) is a small, yet powerful and open source logging library of [Java for mobile devices](#). It is based on the *Log4j* API and is capable of logging to device, to PC or to servers online. It can be used in all phases of application development from debugging on the emulator/device to outdoor field-testing.

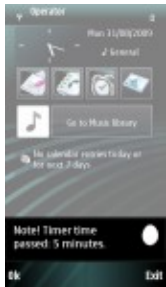
In this article, the author explains the main functionality provided by the Microlog API and how it can be helpful in debugging. The Logger component, Appender component and Formatter component of the API are discussed in detail with apt code examples demonstrating their usage.

[Read the article](#) to gain insight into this very useful logging library.

[How to use SoftNotification API in Java ME](#)

Week 40 - September 27th 2009

This week's featured article, [How to use SoftNotification API in Java ME](#), is written by [Jarmlaht](#).



The *Soft Notification API* has been introduced in the Java Runtime 1.3 for S60 as a new UI feature. It is an extension of Nokia UI API 1.2 and commonly available in the [S60 5th Edition](#) devices.

The soft notifications inform the user of events that have occurred in the user's absence, or while the user was busy with some other applications. They are displayed as pop-ups on the device home screen (idle view). The notification text and the icon image for a notification can be set. It supports several image formats such as png and svg images.

This article explains the usage of *Soft Notification API* using the *EggTimer MIDlet*. The article also contains an example application which is available for download.

[Read the article](#) and start displaying soft notifications in your applications.

[How to drive radio-controlled car with Nokia N97 or Nokia 6260 Slide devices](#)

Week 28 - July 5th 2009

Featured_Articles:_Java

This week's featured article, [How to drive radio-controlled car with Nokia N97 or Nokia 6260 Slide devices](#), is created by [Jarmlaht](#).



The article explains the *MiniController project* which was demonstrated on the Nokia Developer Summit event. The intention was to demonstrate the control of the radio-controlled(RC) car with a mobile device and also to share the latest information about [S60](#) and [Series 40](#) Mobile Java related features.

This article introduces two different solutions for controlling an RC car with a mobile device. The first solution uses the [Nokia N97 device](#), which supports the standard **Mobile Sensor API (JSR-256)**. The second solution uses the [Nokia 6260 slide device](#), where a joystick key is used to control the car. In both the cases the car is controlled by using a small Mobile Java? application that creates and sends the controlling messages to the car.

The main purpose of this article is to describe the software portion of the project; how the controller [MIDlets](#) were created, tested, and used. The readers should have a basic knowledge of [Java](#) programming and how to create [MIDlets](#).

[Read the article](#) to get more insight into the *MiniController project* and control your RC car using the Nokia N97 or Nokia 6260 Slide devices.

Java Porting

Week 27 - June 28th 2009

This week's featured article, [Java Porting](#), is written by [Grahamhughes](#).



[Java ME](#) is well known for the variations in its implementations. While it is one of the most widely available technologies, it is also one of the most fragmented.

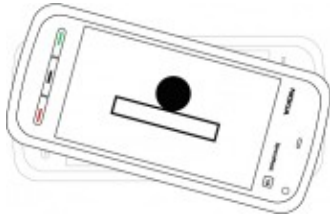
This article aims to identify the most common pitfalls of mobile Java development and provide useful suggestions for writing code that is easy to read, maintain, and port. Code examples are given in order to better illustrate certain points and links to useful resources are also present.

[Read the article](#) to get tips on how to avoid problems related to porting your Java applications and get a general idea of techniques that may help you achieve the *write once, run everywhere* objective.

How to get accelerator sensor values in Java ME

Week 22 - May 24th 2009

This week's featured article, [How to get accelerator sensor values in Java ME](#), is authored by [Jarmlaht](#).



A sensor is any measurement data source. Sensors vary from physical sensors such as magnetometers and accelerometers to virtual sensors, which combine and manipulate the data they have received from other kinds of physical sensors. The featured article focuses on manipulating accelerometer sensor values, thereby the orientation and movement, of devices in J2ME.

The [S60 5th Edition](#) devices support [Mobile Sensor API - JSR-256](#). The API provides means to monitor measured data using [J2ME](#). The application can register a listener and set limits and ranges for monitoring. If the measured value meets any of defined conditions the listener is notified. The API is designed as an Optional Package that can be used with many Java Micro Edition Profiles.

The article contains a code snippet explaining how to search the accelerometer sensor and retrieve sensor values. [Read the article](#) to add accelerometer support to your [J2ME](#) applications and games.

NetBeans

Week 18 - April 26th 2009

This week's featured article, [How to create mobile Java applications using NetBeans](#), is originally created by [Firt](#) and has been recently updated by [Bogdan.galiceanu](#).



The [NetBeans](#) IDE is an open-source integrated development environment based on the NetBeans Platform and has been written entirely in the [Java](#) language. For mobile development, NetBeans provides many Rapid Application Development (RAD) features, such as a Visual Flow Designer, Visual Form Designer, and tools for multitarget development.

To enable the use of NetBeans by developers building mobile Java applications for devices, Nokia has provided support for NetBeans in many of the [Series 40](#) and [S60 SDKs](#), including all the recent Series 40 5th Edition and S60 3rd Edition SDKs.

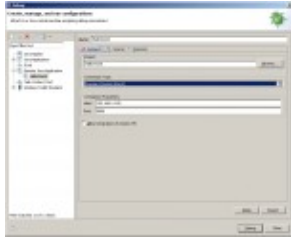
The featured article is a tutorial on getting started with NetBeans. It addresses basics for using the IDE like creating a new Java ME project, creating a flow, building screens, and adding code, building the project and running it in the emulator and deploying the MIDlet to a Nokia device.

[Read the article](#) and which shows how powerful and convenient NetBeans is for creating mobile Java applications.

How to configure Java ODD over WLAN with Eclipse and NetBeans IDEs

**Week 15 - April 5th
2009**

This week's featured article, [How to configure Java ODD over WLAN with Eclipse and NetBeans IDEs](#), is written by [Jarmlaht](#).



ODD (On-device debugging) makes it possible to control the application flow by entering breakpoints and following the variable values in the IDE's debugging view.

This article walks through the steps of setting up the ODD feature for debugging a MIDlet running on a phone by using a data bearer (in this case, WLAN) for sending and receiving data between PC and the phone.

The article explains, how ODD is setup by using the S60 5th Edition SDK 1.0 and the **EcmtAgent tool** included with it. It elaborates how to configure Java ODD over WLAN with Eclipse and NetBeans IDEs with step by step explanation. This feature has been tested with NetBeans 6.5 and Eclipse 3.4 IDEs, also other NetBeans 6.x and Eclipse 3.x versions should work just fine.

Read the article to setup ODD feature and step through the control flow of your Midlet over WLAN.

Java Security Domains

Week 8 - February 15th 2009

This week's featured article, Java Security Domains, is written by Hartti Suomela.



Many Java ME developers are confronted with the problem of signing their MIDlets to be able to access certain functionality and avoid the confirmation messages displayed when the application performs restricted operations.

The article discusses the various aspects one needs to take into consideration before opting for a particular certificate authority and target device platform. In addition, it links to the other articles that provide information about general API access rights on MIDP 2.0 and 2.1, as well as on the various Nokia platform editions, for both untrusted and trusted 3rd party domain. It also provides instructions on how to set the API access settings on S60 and Series 40 devices.

Read the article for a concise yet complete collection of resources that will help you make the best decisions when it comes to signing MIDlets.

Featured_Articles:_Java

This week's featured article is a series of well-designed articles that demonstrate how to develop a game in Java ME by [SergioEstevao](#).

- [How to develop a game - Part 1](#)
- [How to develop a game - Part 2](#)
- [How to develop a game - Part 3](#)
- [How to develop a game - Part 4](#)
- [How to develop a game - Part 5](#)
- [How to develop a game - Part 6](#)
- [How to develop a game - Part 7](#)

How to use Google Maps data in mobile applications

Week 29 - July 13th 2008



Standard Google Maps code is suited for Web applications, since it includes a lot of Ajax functionalities, that are not really useful if you're building a mobile application. The solution is to use the [static maps](#) service which allows retrieving single images, easily usable within a mobile application.

The static maps service supports different image formats (png32, gif, jpg) and customizable image size, so that you can get perfect images for all purposes. Learn how from [Jappit's](#) article [How to use Google Maps data in mobile applications](#).
