

# For a richer mobile experience

## Designing Applications for Smartphones – Series 60 Platform Overview

### Contact Details

---

Nokia Corporation  
Mobile Software Unit  
P.O. Box 100  
FIN-00045 NOKIA GROUP,  
Finland  
Tel. +358 (0) 7180 08000  
[www.nokia.com](http://www.nokia.com)

The contents of this document are copyright ©2002 Nokia. All rights reserved. A license is hereby granted to download and print a copy of this document for personal use only. No other license to any other intellectual property rights is granted herein. Unless expressly permitted herein, reproduction, transfer, distribution or storage of part or all of the contents in any form without the prior written permission of Nokia is prohibited.

The content of this document is provided "as is", without warranties of any kind with regards its accuracy or reliability, and specifically excluding all implied warranties, for example of merchantability, fitness for purpose, title and non-infringement. In no event shall Nokia be liable for any special, indirect or consequential damages, or any damages whatsoever resulting from loss of use, data or profits, arising out of or in connection with the use of the document. Nokia reserves the right to revise the document or withdraw it at any time without prior notice.

Nokia and Nokia Connecting People are registered trademarks of Nokia Corporation. Nokia product names are either trademarks or registered trademarks of Nokia. Other product and company names mentioned herein may be trademarks or trade names of their respective owners.

**TABLE OF CONTENTS**

<b>1. Executive Summary</b>	<b>3</b>
<b>2. Overview</b>	<b>4</b>
2.1 Reference Applications	4
2.2 Platform Architecture	6
2.3 Symbian OS	6
2.4 User Interface	7
2.5 Communications	8
<b>3. Developing applications</b>	<b>9</b>
3.1 Java Applications	9
3.2 Native Symbian Applications	10
3.3 Messaging	12
3.4 Browsing	12
<b>4. Summary of Technologies</b>	<b>14</b>
<b>5. Resources</b>	<b>15</b>

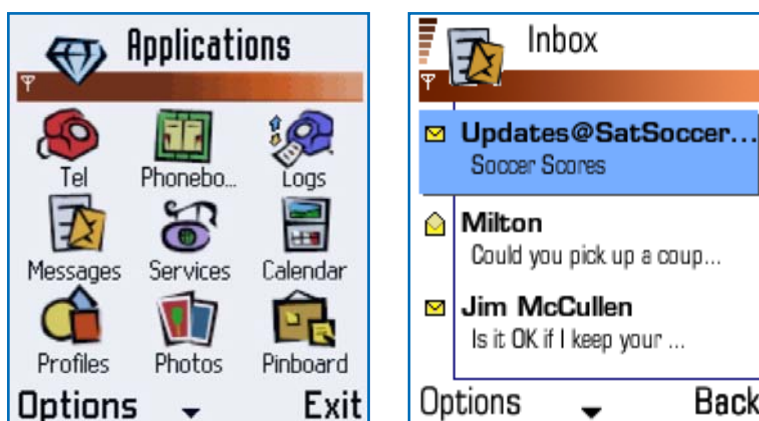
## 1. EXECUTIVE SUMMARY

The introduction of smartphones is opening up new opportunities for software developers. The new, open software platforms empower developers to design innovative mobile applications and services in a rapidly growing market.

The Series 60 Platform, with its associated development tools, is set to grow this smartphone market. It includes a graphical user interface and a suite of applications all built upon the Symbian operating system. Software Development Kits for Series 60 facilitate the creation of applications in Java, or C++, downloading the applications, and running these same applications. The Platform enables software developers, operators and device manufacturers to create their own solutions – solutions that will be interoperable across different devices and networks.

Technology intelligence firm IDC forecasts that 64 million smartphones will be shipped to customers in year 2005. It will be more than all other handheld devices, PDAs and industrial handheld devices together. By 2005, IDC expects Symbian OS (EPOC) to have the largest market share in smartphone operating systems.

Figure 1. A design for the Series 60 Platform user interface



Series 60 Platform draws its design and functionality from the experience of Symbian (jointly established by Ericsson, Matsushita, Motorola, Nokia, Psion) and Nokia. Putting an operating system designed from the beginning for small mobile devices together with the most popular user interface components of the industry, add up to the environment where applications will thrive.

As a standards based platform, Series 60 and applications developed for it can run on any participating manufacturers' hardware, creating a new growing market for the software industry and a safe platform investment for a software vendor.

The mobile world is now open for all innovators who envision a future driven by applications and services – on an open architecture globally shared between software companies, content publishers, mobile operators and device manufacturers. Join the community at [www.forum.nokia.com](http://www.forum.nokia.com).

## 2. OVERVIEW

New technologies, services, types of mobile devices and the things they can do are constantly moving forward. Manufacturers are all trying different approaches, with screen size, keypads, browsers and other elements all being varied, in the search for a product that most appropriately meets the customers needs. Applications, services and content must all be adapted to different devices much like traditional content is adapted for newspapers, magazines, radio and television.

Industry leaders have committed to create a global and open mobile architecture for interoperable applications, content and services. The scope of this industry initiative encompasses both terminal client software modules for mobile terminal vendors, and corresponding server solutions for mobile operators. Doing this will offer mobile operators, systems integrators, IT suppliers, terminal manufacturers, and application developers new avenues of growth and revenue by enabling a multi-vendor ecosystem, built on open industry standards such as WAP2.0/XHTML, MMS (Multimedia Messaging Service), SyncML and other industry standard technologies.

An application developer who wants to reach the widest possible market, wants to adhere to standards and to use popular programming languages, like Java and C++. The crucial standards and development tools, amongst many other protocols and utilities, have been implemented in Series 60 Platform, providing smartphones with an open-architecture based operating environment.

In mobile devices, the screen and keyboard hardware dictate how the user interface portion of the software is designed. In order to promote interoperability from the start, Series 60 Platform defines a common user interface for smartphones that are one-hand operated and have a 176 x 208 screen size. This creates a uniform user interface and screen format to which developers can write, therefore enabling simplification and savings on development cost.

Application designers can use Series 60 SDK (Software Development Toolkit) and create both stand-alone and server-connected applications for smartphones. Also, separate Java and WAP/XHTML toolkits can be used for designing applications for Series 60. The applications developed once for Series 60 Platform will then run on other Series 60 Platform based smartphones.

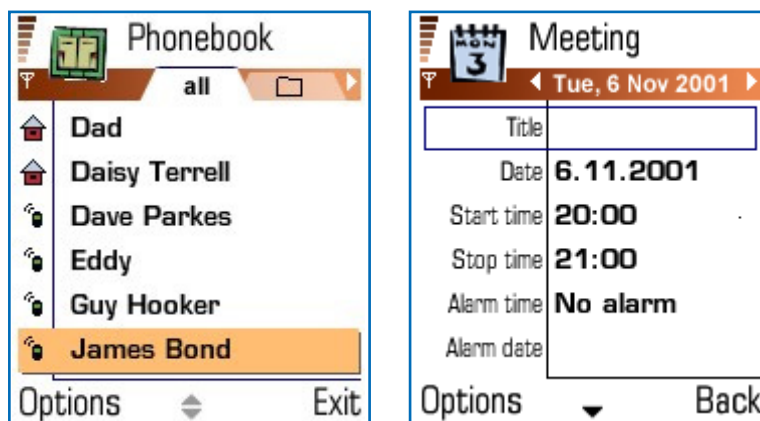
In addition to mobile devices, developers can find opportunities in designing software products for servers. For instance, mobile network operators and service providers provide content and application services for their subscribers. Enterprises need software that connects their mobile employees to the Intranet and applications that can be used for customer service and mobile commerce.

### 2.1 Reference Applications

The Series 60 Platform includes ready-to-run applications which hardware manufacturers can embed in their Series 60 devices. Not only these applications illustrate the possibilities of the platform, but they also guide developers in designing software that complies with the user interface style.

Reference applications provide public APIs (Application Programming Interfaces) for accessing their services from other applications. For example, the Phonebook application has a service for displaying a list of contacts, the Photo Album application has a service for finding images, and the Messages application a service for sending emails.

Figure 2. Some of the applications in the Series 60 Platform: Phonebook and Calendar.



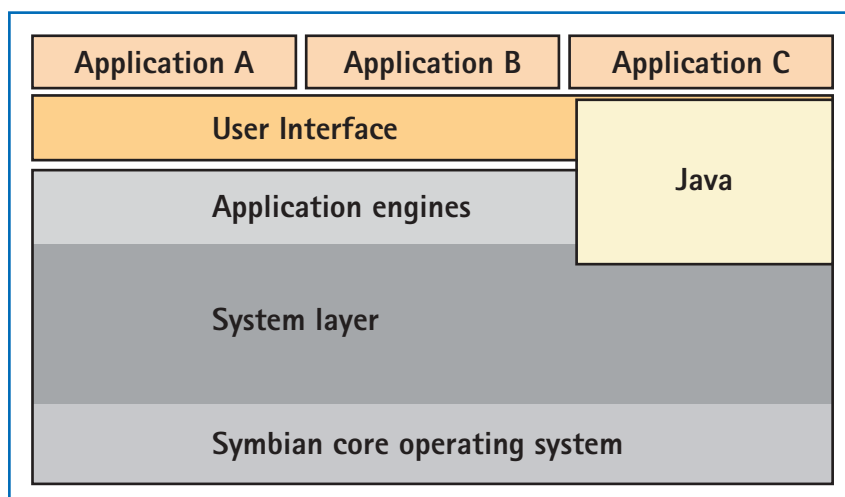
Phonebook	Contact database integrated with messaging and other applications. Supports vCard.
Calendar	Scheduling application, supports vCalendar data format.
To-Do List	vCalendar items can be downloaded into this to-do list.
Notepad	For text entries.
Photo Album	Store for multimedia messages and images.
Pinboard	Application management tool for the multitasking environment.
Clock, Calculator, Games, Unit Converter	World clock, business calculator and a selection of entertainment applications.
Composer	Lets users compose new tunes for applications.
Email, SMS and MMS	Client software for messaging applications.
Telephony applications	Voice Recorder, Telephone Settings, Call Logs & Message Indicators, User Profiles, Call Forwarding (Divert), Speed Dialing and Voice Dialing & Voice Tags.
Application installation	Installing new software via the PC Connectivity suite.
Synchronization	SyncML 1.0.1 synchronization engine. Supports data transfer over mobile networks (WAP), Bluetooth and infrared (IrDA). vCal 1.0 and vCard 2.1 data formats.
Security	Security settings and software certificate management.

## 2.2 Platform Architecture

Series 60 Platform builds on the Symbian operating system, complementing it with a graphical user interface library and reference applications.

Symbian OS components provide data management, communications, graphics, multimedia, security, application engines, messaging engine, Bluetooth, browser engines and support for data synchronization and internationalization.

Figure 3: Series 60 Platform Architecture



The kernel, file server, memory management and device drivers are located in the Symbian core operating system layer.

The system layer provides communication and computing services, such as TCP/IP, IMAP4, SMS, and database management.

The application engines enable software developers to create their own user interfaces to the application data and databases. Data synchronization is provided through the PC Connectivity suite.

The user interface software created on top of the Symbian OS is adapted for a particular device category, in this case smartphones. The ready-to-run and custom developed applications make up a fully functional mobile device.

The Java runtime environment allows the download and deployment of Java MIDP applications in Series 60 devices.

## 2.3 Symbian OS

Symbian operating system derives from the EPOC platform originally developed by Psion for small handheld devices. Today the operating system is developed by Symbian, the company established by Ericsson, Matsushita, Motorola, Nokia and Psion.

Symbian OS is a 32-bit multitasking operating system, where applications are designed to interact with one another. For example, if a phone call interrupts a user of composing an email message, the user can switch from email to a calendar application in the middle of a telephone

conversation. Or, an incoming SMS may trigger the user to access the contact database and forward the SMS on.

By complying with the platform architecture and software design guidelines, application designers can routinely manage such occurrences in the daily lives of smartphone users.

From the start, Symbian was designed for small battery-powered devices with wireless communications. The design cornerstones of the Symbian OS have been:

- Performance – the Symbian OS is designed to make minimal demands on batteries and to have low memory consumption.
- Multi-tasking – telephony and universal messaging are fundamental components. All applications are designed to work seamlessly in parallel.
- Standards – the use of technologies based on agreed standards is a basic principle of Symbian, ensuring that applications are portable and interoperable.
- Object-oriented software architecture.
- Memory management optimized for embedded software environment.
- Security mechanisms for enabling secure communications and safe data store.
- Application support for international environment with built-in Unicode character sets.

## 2.4 User Interface

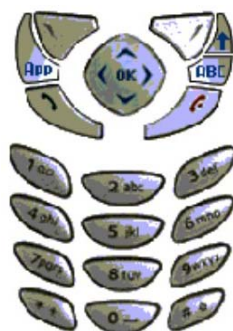
A smartphone is primarily a mobile phone, a fact reflected in the user interface optimized for one-handed operation. But a smartphone is also a handheld device for diverse applications and data communication. Fitting these requirements together into a graphical user interface that anyone, who can use a mobile phone, can master, makes a difference between smartphones and other handheld devices. This is a crucial distinction compared to PDAs (Personal Digital Assistants), which require a pen or a keyboard and the use of both hands for operating them.

The user interface design of an application for the Series 60 Platform typically starts from dividing the structure into browsing elements and detailed views. For example, the browsing view allows the user to select one entry from a list of elements. When the user activates an entry, the respective detailed view displays the data.

A wide selection of user interface elements are available for a developer ranging from list boxes, standard dialogs, pop-up menus, check boxes, radio buttons to rich text and graphics.

Series 60 Platform supports a 176 x 208 pixel screen size. The platform specifies a 12-key numeric keypad with function keys. Function keys include two softkeys, 4-way navigation key, an application launch-and-swap key, and 'send' and 'end' keys. For improving text input, the keyboard also specifies a clear key and an alpha toggle key.

Figure 4. An illustration of a keypad with Series 60-supported keys.



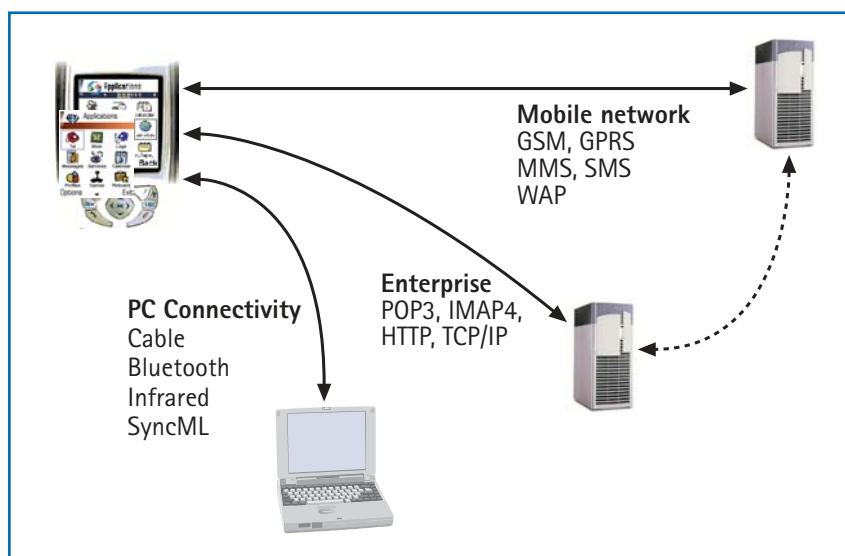
## 2.5 Communications

Application designers who create solutions that communicate with other systems, require standards for ensuring interoperability. Series 60 Platform relies on standard protocols that have been defined for communication, messaging, browsing and data synchronization. For example, a Java application in a mobile phone, which extracts data from a corporate customer database may use HTTP for communication. A game can be downloaded into a phone transferring it over the WAP protocol through a mobile network. Synchronization of a contact list may take place at the user's PC using SyncML over Bluetooth.

In addition to stand-alone applications designed for Series 60 devices, server applications, server utilities and middleware software represent a market for software companies. Server applications may be hosted by a mobile network operator who provides them as a service to subscribers. Mobile portals, payment solutions, location based solutions, advertising and virtual communities are examples of these services.

For enterprises, GPRS means IP connectivity to mobile devices through mobile networks. Server software for mobile applications can reside in corporate networks allowing Intranet applications to reach employees out of the office, and facilitating mobile commerce applications to reach customers.

Figure 5. Series 60 Platform Communication Technologies.



Data transfer between applications in different devices is enabled by standards that define syntax and format for data. Series 60 Platform features a SyncML engine, which can be accessed through a public API. PC Connectivity over wireless or cable connection can be used for data synchronization and application installation.



### 3. DEVELOPING APPLICATIONS

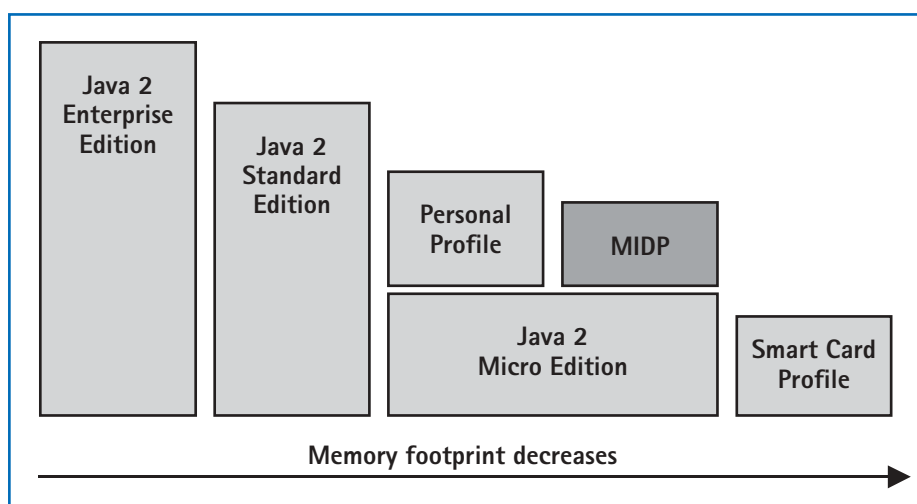
Application developers from computer, networking and telecommunications worlds are looking at smartphones as a new platform paradigm. However, developers can still use familiar programming tools together with new toolkits for turning their ideas into innovative new applications. Series 60 Platform has been packaged as a Software Development Kit (SDK) for developers. The SDK features the APIs, documentation, and an emulator to allow testing of the software before downloading into an actual device.

Depending on the intended use of the application software, it can be designed as a Java application, as a Symbian C++ application, a messaging application, or a content browsing application.

#### 3.1 Java Applications

Java has become popular as a programming language and as a run time environment ranging from small devices to network servers. In their book 'Programming Wireless Devices with the Java 2 Platform, Micro Edition', Roger Riggs, Antero Taivalsaari, Mark VandenBrink estimate that there are more than 2.5 million Java developers worldwide. Nokia intends to deliver more than 50 million mobile terminals supporting the Java platform by the end of 2002.

Figure 6. Java Technology Platforms.



Series 60 Platform accommodates the J2ME environment according to the MIDP profile of the CLDC configuration definition. The CLDC is designed to be a small-footprint configuration for resource-constrained devices. MIDP is a functional extension to the CLDC, defining the profile for wireless information devices, such as smartphones. Midlets, MIDP applications, can be downloaded from the internet over the air and executed in smartphones.

For application developers, Java gives hardware independence, the support of a large development community, and built-in security. Java frees developers from dealing with the specifics of different devices allowing them to concentrate on the applications.

Examples of downloadable applications:

- Interactive information applications such as maps and city guides, corporate information tools.
- Entertainment applications, for example games, animated characters and virtual pets.
- Financial applications, such as banking and stock trading.

In a smartphone environment, Java is best used for applications where time to market is critical, the client application acts as an extension to a server application, application requirements update frequently, and deployment of the application to different platforms is important.

Series 60 smartphones can be connected to the Internet, for instance using packet-based GPRS mobile networks. Network and PC connectivity allows users to download the applications and data of their choice into their smartphones. Since all users may not be security aware, or verifying the origins of the downloadable software may be difficult for users, Java has a built-in sandbox security model to protect the system from malicious software. The security is achieved by setting clear boundaries what applications can do in the device.

Since Java has been designed to run on any device with a Java virtual machine, there are some compromises in its generic functionality. A Java application can't access all features of a device, such as the contact database or the calendar – the full functionality is available in the C++ development environment.

### 3.2 Native Symbian Applications

From the start, Symbian operating system has been designed specifically for small handheld devices, rather than trying to downscale a PC operating system to fit into an embedded software environment. Symbian OS features power management built within the kernel, object-oriented architecture, sophisticated memory management, event handling mechanisms, and multitasking.

C++ is the native programming language of Symbian OS, so all Series 60 Platform API calls are available for a C++ programmer.

A smartphone application may be kept running for weeks, making memory management critical for the operating system. Symbian provides programming methods, such as heap checking, (debug) asserts, naming convention and leave-trap mechanisms, which enforce good memory management.

Figure 7. Overview of Series 60 APIs.

Application Programming Interface	Function
Application Engines	Access to the data of the service applications, like Calendar, To-Do list, Contacts etc.
Application Framework	Core frameworks and libraries for application software.
Application Services	Utility services for applications, including alarms, logging, system information, and vCard and vCalendar handling.
Communication	Frameworks and system services for communications, such as sockets.
Infrared	Infrared communications.
Messaging	Framework for multi-protocol messaging, for example email, MMS and SMS.
Multi Media Server	Music and image manipulation and format conversion. Supports graphics, animation, audio files and video clips.
Networking	TCP/IP and dial-up networking services.
PC Connectivity	Backup, restore, file format conversions, application installation and printing.
Serial Communication	Framework for serial communications services.
System	System functions, such as security, clipboard, date and time, database management, file server, graphics, locale settings, memory management, thread and process management, timers.
Telephony	Interface for initiating, controlling, and terminating telephone calls.
User Interface	User interface library for Series 60.
WAP Stack	Access to the WAP protocol stack.

### 3.3 Messaging

Messaging is ideal for mobile communication applications that don't require an immediate response from a server. A client application puts together the message, sends it away and lets the mobile network store-and-forward system take care of the rest. Messaging also frees the user for other tasks, while waiting for a response.

The success of SMS has proved how beneficial it is for the whole industry when an application is available on mobile phones from different manufacturers, and across different operator networks. Mobile messaging is now evolving from SMS text messaging to Multimedia Messaging Service (MMS). Practically all existing, commercially successful SMS services can also be implemented using MMS. New infotainment, chatting, instant messaging, and person-to-person messaging services are opportunities for developers who understand consumers' demand for instant creation and consumption of content.

In the corporate world, email is a vital tool for business people, who can now access their email and other messaging-enabled corporate applications with their smartphones. For example, a database application that keeps users up-to-date via email, can communicate with business users who have a secure connection from the smartphone to the corporate Intranet.

Developers can make full use of the messaging facilities in Series 60 Platform. Messaging offers plenty of opportunities to create messaging enabled client applications, and to create plug-in modules to support customized messaging types. The built-in messaging framework in Series 60 provides an API for sending and receiving SMS, MMS, and email (IMAP4, POP3, SMTP).

A useful feature for programmers is the Send-as API that helps create outgoing messages in client applications. It provides a common method of composing messages, no matter what type and transport will be used, and features a simple user interface module for sending the message on its way.

### 3.4 Browsing

It is well understood that browsing on a mobile device is very different from browsing on a personal computer. A mobile device must show essential, personally relevant information, in a compact and visually appealing manner.

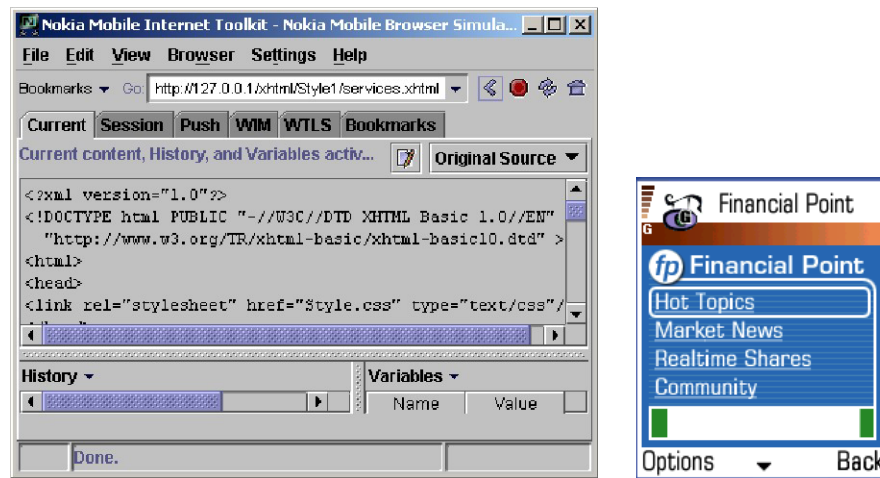
The Wireless Application Protocol (WAP) is a global standard for mobile Internet applications and browsing. Functionally similar to the World Wide Web, it is designed to accommodate the limited memory and small screens of mobile devices, which are connected to servers over low bandwidth connections. Packet-based GPRS networks make accessing WAP pages and applications more convenient over Circuit Switched Data (CSD) connections by shortening connection times and increasing the data transfer speed.

Examples of applications that involve real-time interaction or long lasting sessions are interactive games, on-line auctions, chat, and newsgroups. A banking transaction on a smartphone is an example of an application that requires secure, online connection between the client and the server all the way through the transaction. A secure session over a browser connection is a proven way to implement such an application.

Series 60 Platform supports WAP 1.2.1 / WML browsing. Features include push functionality, over the air configuration of WAP settings, content download, WTAI, and WTLS security.

Nokia has made available a Mobile Internet Toolkit product for application designers and content publishers to create both XHTML and WML content. The Toolkit is available for download at [www.forum.nokia.com](http://www.forum.nokia.com).

*Figure 8. Mobile Internet Toolkit: scripting and application screens.*



Mobile Internet Toolkit runs on a PC, providing a complete scripting, testing and simulation environment for developers. The included emulator is adjustable for different screen sizes, including 176 x 208 used in the Series 60 Platform. Developers can consequently create content and applications without an XHTML device, or without access to a mobile network operator infrastructure.

#### 4. SUMMARY OF TECHNOLOGIES

Bluetooth		A specification for short-range wireless communications.
C++		Object-oriented programming language.
CLDC	Connected Limited Device Configuration	Specification of Java environment for consumer devices with limited resources.
CSD	Circuit Switched Data	Circuit switched connection for transferring data over a mobile network.
CSS	Cascading Style Sheets	Content formatting and rendering technology for XHTML.
GPRS	General Packet Radio System	A radio technology for GSM networks. Transmission rates with theoretical maximum of 171.2 Kbps.
GSM	Global System for Mobile communications	Digital mobile phone system.
HSCSD	High Speed Circuit Switched Data	Data communications technology for GSM. Transmission rates between 14.4Kbps and 57.6 Kbps.
HTTP	Hyper Text Transfer Protocol	Data transfer protocol.
IMAP4	Internet Message Access Protocol	Email protocol.
IrDA	Infrared communication	A suite of protocols for exchanging data via infrared.
Java		Industry standard object-oriented language and run-time execution environment.
J2ME	Java 2 Micro Edition	Java edition for consumer devices.
MIDP	Mobile Information Device Profile	An extension API to the CLDC, which addresses the specific needs of the mobile phone market.
MMS	Multimedia Messaging Service	Protocol defined by 3GPP. Messaging for text, images, audio.
POP3	Post Office Protocol	Email protocol.
SMS	Short Message Service	Text messaging service in GSM networks.
SMTP	Simple Mail Transfer Protocol	Email protocol.

Bluetooth		A specification for short-range wireless communications.
SyncML		Industry standard for universal synchronization of remote data and personal information.
TCP/IP	Transmission Control Protocol / Internet Protocol	The Internet protocol.
WAP	Wireless Application Protocol	Protocol optimised for viewing content over mobile networks.
WML	Wireless Markup Language	XML based markup language optimised for mobile devices with small screen.
WTAI	Wireless Telephony Application Interface	WAP tools for creating telephony applications.
WTLS	Wireless Transport Layer Security	The WTLS layer provides privacy, data integrity and authentication for WAP.
vCalendar		Data format for exchanging calendaring and scheduling information.
vCard		Electronic business card data format.
XHTML	Extensible HTML	XML-based markup language.

## 5. RESOURCES

More specific information on Series 60 Platform, additional information about specific technologies, toolkit downloads and a developer network is all available at Forum Nokia:

[www.forum.nokia.com](http://www.forum.nokia.com)

Symbian information:

[www.symbian.com](http://www.symbian.com)

Java information and downloads:

[java.sun.com](http://java.sun.com)

XHTML and other Internet protocols:

[www.w3c.org](http://www.w3c.org)

WAP information:

[www.wapforum.org](http://www.wapforum.org)

MMS, SMS and other telecommunication standards:

[www.3gpp.org](http://www.3gpp.org)

Information about SyncML:

[www.syncml.org](http://www.syncml.org)

vCalendar and vCard:

[www.imc.org](http://www.imc.org)