

---

# S60 3rd Edition: What's New for Developers

**Version 1.7**  
June 6, 2007

S60 platform

## Legal notice

Copyright © 2005, 2006, 2007 Nokia Corporation. All rights reserved.

Nokia, Nokia Connecting People, and Nokia Eseries are trademarks or registered trademarks of Nokia Corporation. Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. Bluetooth is a registered trademark of Bluetooth SIG, Inc. Other product and company names mentioned herein may be trademarks or trade names of their respective owners.

### Disclaimer

The information in this document is provided “as is,” with no warranties whatsoever, including any warranty of merchantability, fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification, or sample. This document is provided for informational purposes only.

Nokia Corporation disclaims all liability, including liability for infringement of any proprietary rights, relating to implementation of information presented in this document. Nokia Corporation does not warrant or represent that such use will not infringe such rights.

Nokia Corporation retains the right to make changes to this specification at any time, without notice.

### License

A license is hereby granted to download and print a copy of this specification for personal use only. No other license to any other intellectual property rights is granted herein.

## Contents

<b>1.</b>	<b>Introduction .....</b>	<b>6</b>
1.1	Purpose and scope .....	6
<b>2.</b>	<b>General developer issues.....</b>	<b>7</b>
2.1	ABI for the ARM® Architecture .....	7
2.2	Real-time kernel.....	7
2.3	Platform security .....	7
2.3.1	Trusted computing base .....	8
2.3.2	Data caging .....	8
2.3.3	Capability model .....	8
2.4	User interface .....	9
2.5	Tools.....	9
2.6	Costs and benefits .....	9
2.6.1	Certification procedures.....	9
2.6.2	Developer benefits.....	10
2.7	Code base maintenance and future-proofing.....	11
<b>3.</b>	<b>APIs and other changes .....</b>	<b>12</b>
3.1	C++ APIs in S60 3rd Edition .....	12
3.1.1	Location APIs .....	12
3.1.2	MMF DRM API .....	12
3.1.3	Public SIP API and SDK.....	13
3.1.4	Light API .....	13
3.1.5	Web services APIs .....	13
3.1.6	IM APIs .....	13
3.1.7	Exif API.....	13
3.1.8	Find Item API.....	13
3.2	C++ APIs in S60 3rd Edition, Feature Pack 1.....	13
3.2.1	Gallery Content Listing API .....	13
3.2.2	Optical Character Recognition API.....	13
3.2.3	Central Repository Notification Handler API .....	13
3.2.4	Profiles Engine API.....	13
3.2.5	Screen Mode API .....	14
3.2.6	OpenGL v1.1 API .....	14
3.2.7	Browser Control API .....	14
3.2.8	Telnet Engine API.....	14
3.2.9	App Framework Animation API .....	14
3.2.10	OBEX API (updated) .....	14
3.2.11	OBEX MTM API (updated) .....	14

3.2.12	SIP APIs .....	14
3.3	C++ APIs in S60 3rd Edition, Feature Pack 2.....	15
3.3.1	RGA APIs .....	15
3.3.2	AIW framework APIs .....	15
3.3.3	Map framework APIs .....	15
3.3.4	Application network-level roaming APIs.....	16
3.3.5	Middle Softkey API .....	16
3.3.6	Status Pane API .....	16
3.3.7	Other updated APIs.....	16
3.4	Open C.....	16
3.5	Java™ APIs in S60 3rd Edition.....	17
3.5.1	Location API for J2ME™ (JSR-179).....	17
3.5.2	Security and Trust Services API for J2ME™ (JSR-177) .....	17
3.5.3	SIP API for J2ME™ (JSR-180).....	17
3.5.4	Wireless Messaging API 2.0 (JSR-205).....	17
3.6	Java™ APIs in S60 3rd Edition, Feature Pack 1 .....	17
3.6.1	Advanced Multimedia Supplements API (JSR-234).....	17
3.6.2	Scalable Vector Graphics 2D API (JSR-226).....	17
3.6.3	Mobile 3D Graphics API for J2ME™ v1.1 (JSR-184).....	17
3.7	Java™ APIs in S60 3rd Edition, Feature Pack 2 .....	17
3.7.1	Mobile Information Device Profile 2.0 (JSR-118).....	18
3.7.2	Mobile Service Architecture (JSR-248), subset.....	18
3.7.3	Scalable 2D Vector Graphics API for J2ME™ (JSR-226).....	18
3.7.4	Java™ APIs for Bluetooth (JSR-82).....	18
3.8	S60 customization framework.....	18
3.9	Bluetooth standard v1.2 support .....	18
3.10	OMA standards support .....	19
3.10.1	OMA DRM 2.0: base services .....	19
3.10.2	OMA SyncML DS v1.2.....	19
3.11	Interoperability with PCs and Microsoft Exchange Server.....	19
3.12	Flash Lite from Adobe .....	19
3.13	Web widgets.....	19
3.14	API documentation.....	20
<b>4.</b>	<b>Conclusion.....</b>	<b>21</b>
<b>5.</b>	<b>Terms and abbreviations.....</b>	<b>22</b>
<b>6.</b>	<b>Evaluate this resource .....</b>	<b>23</b>

## Change history

February 14, 2005	Version 1.0	Initial document release
November 9, 2005	Version 1.1	Information on S60 3rd Edition, Feature Pack 1 added
December 16, 2005	Version 1.2	Updated DRM information in Chapter 3
December 23, 2005	Version 1.3	Updated JSR support details in Chapter 3
July 3, 2006	Version 1.4	Updated details of the content of S60 3rd Edition, Feature Pack 1 in Chapter 3
February 7, 2007	Version 1.5	Updated details of the content of S60 3rd Edition, Feature Pack 2 and information on Open C in Chapters 2 and 3, respectively
April 18, 2007	Version 1.6	Added details on RGA APIs to Chapter 3
June 6, 2007	Version 1.7	Added details on Web Run-Time to Chapter 3

---

# **1. Introduction**

This document examines the changes introduced in S60 3rd Edition that affect application development. It also discusses the range of new APIs and other features added to the platform.

## **1.1 Purpose and scope**

This document is aimed at developers who are about to create applications for S60 3rd Edition or who want to port an existing application to this latest version of the S60 platform.

---

## 2. General developer issues

Some key changes in S60 3rd Edition alter the way applications are developed. These changes include a new binary, a new kernel, and the implementation of Symbian OS platform security.

### 2.1 ABI for the ARM® Architecture

From S60 3rd Edition onward, the platform is based on the Application Binary Interface (ABI) for the ARM® Architecture, and a new compiler is used for applications. The ARM C/C++ ABI is an industry standard that determines how executables and shared objects work together. It was developed by ARM, together with other operating-system and software-tool vendors, and it is optimized to meet the unique needs of embedded-application development.

Via the ARM ABI standard, the S60 platform offers improved performance for system and third-party applications through increased efficiency in memory usage and data access time. The standard also allows interoperability among software components from different vendors; developers can now create compliant libraries for the ARM Architecture, using any ARM ABI-based compiler.

### 2.2 Real-time kernel

The underlying operating system for S60 3rd Edition is Symbian OS v9.x, which features a new real-time kernel, the EPOC Kernel Architecture 2 (EKA2). (EPOC was the original name for Symbian OS.) The previous kernel was EKA1. The new multithreaded kernel has predictable execution times for many kernel services and reduced latency times, so it is ideal for time-critical applications such as communications and IP stacks; multimedia applications that require high bandwidth, such as video streaming; and voice over IP (VoIP).

Development of the EKA2 kernel has resulted in improved emulation. The Win32 emulator for EKA2 shares the same core kernel code with device implementations of the kernel, ensuring a more faithful emulation of the target hardware; in particular, process emulation is supported. This enhancement means a reduction in development time for some projects, because less on-device debugging is required. It should also mean a reduced need to write WINS platform-specific code.

The EKA2 kernel allows S60 devices to be built on a single-processor architecture, potentially reducing the cost of materials for an S60 device. A reduction in device costs will enable S60 manufacturers to create phones in the midrange, enhancing the potential for shipping significantly higher volumes of devices. For developers, this means a significantly larger market for applications.

### 2.3 Platform security

S60 3rd Edition brings a new approach to platform security. The changes were motivated by the goal of ensuring the integrity of S60 devices so that consumers can be confident that their devices are secure, reliable, and predictable. To achieve this, access to sensitive data and device operations is controlled.

The changes mean that developers have, with the right certification, access to a greater range of APIs than before. The changes also provide some protection from unwanted side effects caused by defects in applications. From a business point of view, the enhancements improve the S60 platform's reputation for stability and security. Platform

security is fully supported in the emulator, thereby allowing developers to test how this feature will affect their applications.

Note that *platform security* does not refer to those security features that were already available in the S60 platform or that are available as third-party extensions to the platform, such as encryption, firewalls, Java™ MIDP security domain model, virus protection, or secure versions of communications protocols.

### 2.3.1 Trusted computing base

To enforce the security measures introduced in S60 3rd Edition, a collection of software known as the trusted computing base (TCB) is used. The TCB contains the kernel, the file system, and the software installer and is responsible for ensuring that only applications with the necessary permissions and authority can be installed and allowed to access restricted areas of the device.

### 2.3.2 Data caging

The goal of data caging is to control access to the file system so that data remain secure. Each application has its own private directory for its stored data that cannot be accessed by other applications. An application can access most areas of the file system outside its private directory, but there are limitations. Without an agreement with the device manufacturer, the `/sys` directory, which stores executable files, will be inaccessible, and the `/resource` directory, which contains application resources such as bitmaps, will be read-only.

### 2.3.3 Capability model

The purpose of the capability model is to ensure that only certified applications are able to use certain APIs and system resources. A device's end user (that is, the person who installs an application) can grant certain permissions, such as the ability to send a short message service (SMS) message and to read and write user data. However, several other capabilities will be available only to Symbian Signed applications.

There are four sets of capabilities.

- *Open to all* — These capabilities are open to all applications; applications do not need to be Symbian Signed to access these capabilities. They comprise about 60 percent of the available APIs.
- *Granted by the user at installation time* — These capabilities include access to Contacts and Calendar entries and use of local and wide area network services. Access to these capabilities is granted by the user to self-signed applications when they are installed on a device. Symbian Signed applications can access all these capabilities and do not require the user to grant permission.
- *Granted through Symbian Signed* — These capabilities include the *Granted by the user at installation time* capabilities and an extended set of features that require that an application be Symbian Signed before the capabilities can be accessed.
- *Granted by the manufacturer* — These capabilities require an agreement with the device manufacturer to allow the capabilities to be granted during the Symbian Signed process.

For more details on the device features associated with each capability, see the document titled *Symbian OS: Overview to Security*, available from the Forum

Nokia Web site

(<http://www.forum.nokia.com/main/resources/technologies/symbian/documentation/networking.html>).

## 2.4 User interface

The S60 platform's UI style has been consistently maintained through the evolution of the S60 platform. Key functions have remained similar, and developers have been able to build to a consistent interaction paradigm.

The introduction of S60 3rd Edition, Feature Pack 2 has brought the first significant change to the S60 UI interaction mechanism: the introduction of a middle softkey. The additional softkey enables improved access to application features by offering more opportunities for function access.

## 2.5 Tools

The use of the ABI binary for S60 3rd Edition means that developers need to obtain new compiler tools that conform to the ARM C/C++ ABI standard. Developers are able to choose from any conformant tools.

The first to support this standard was the ARM RealView<sup>®</sup> RVCT compiler, targeted primarily at S60 licensees and embedded-software developers. This compiler is a powerful set of tools designed to optimize utilization of resources available on the target architecture. The RealView tools are designed and extensively tested by engineers who created the ARM Architecture, making them a robust development solution for systems based on ARM technology. RVCT is suitable for ROM builds and enables significant memory and power savings.

For independent developers who might want low-cost tools, the GNU Compiler Collection (GCC) ABI is available free with the S60 3rd Edition SDK.

The Carbide.c++ tools from Nokia provide support for ARM C/C++ ABI-standard compilers.

## 2.6 Costs and benefits

This section weighs the initial outlay required for developing on S60 3rd Edition against the considerable benefits and enhancements that this platform provides.

### 2.6.1 Certification procedures

There are two main sources of additional costs for developers wishing to create applications for S60 3rd Edition: the new tools that are required for development (see Section 2.5, "Tools") and the certification procedures involved in obtaining Symbian Signed status for an application.

With respect to certification, there are three stages in the development process. These, along with their associated costs, are:

- *Emulator development* — This stage incurs no extra costs for developers beyond those associated with tools changes.
- *Device testing* — To enable testing on devices, developers must obtain Symbian Developer Certificates. These are available free of charge (although a VeriSign Authenticated Content Signing [ACS] Publisher ID, which is also

required for Symbian Signed, is required to obtain certificates for more than one device).

- *Symbian Signed* — Symbian Signed status is granted to an application by a test house for a fee. Current testing prices can be found on the Symbian Signed Web site (<https://www.symbiansigned.com/app/page/overview/testhouses>).

Symbian Signed is recommended for all S60 3rd Edition applications, though it is not necessary for applications using APIs from the *open to all* and *granted by the user at installation time* capabilities. Symbian Signed offers several advantages to S60 3rd Edition applications:

- *Greater channel access* — The additional assurance that Symbian Signed provides as to the quality of an application means that more retail channels are likely to accept the application into their catalogs.
- *Simplified installation* — Symbian Signed removes the user installation warning and the need to grant permission to use *granted by the user at installation time* capabilities, making it more likely that the user will install the application.
- *Improved protection of intellectual property rights* — A Symbian Signed \*.sis file offers several mechanisms that prevent unauthorized repackaging and distribution of applications and content.

## 2.6.2 Developer benefits

Although creating applications for S60 3rd Edition may entail additional costs, for most developers the benefits of the changes in the platform outweigh those costs.

- *Application data security* — The platform now provides a better mechanism for ensuring that application data are maintained in a private, compartmentalized area that can be accessed only by the application itself. This provides enhanced security for enterprise and other data-sensitive applications.
- *Application efficiency* — With its real-time capabilities, the platform has seen enhanced application performance and allowed for the development of real-time applications. This, combined with enhanced support for localization, provides the potential for real-time location-based information services.
- *Wider access to S60 APIs* — Because of the enhanced requirements for certification, Nokia has been able to extend the range of APIs that are open to developing parties. There are many APIs that are now publicly accessible so that the functionality can be directly accessed, allowing enhancements to all types of applications.
- *Increased market confidence* — Certification, effectively a requirement for complex applications, promotes consumer and distributor confidence that all such applications will work efficiently, safely, and correctly. Because the market is still in a growth phase, issues of trust and reliability are important in determining which devices and platforms will be the dominant players. As with so many other aspects of smartphone development, Nokia has taken the lead. The enhancements Nokia is providing promote its reputation as a key innovator in the market. Clearly, it benefits development companies to take advantage of this situation as soon as possible.

## 2.7 Code base maintenance and future-proofing

S60 3rd Edition includes the scalable UI APIs and feature discovery APIs that enable developers to maintain their S60 code bases and provide some future-proofing for applications.

The scalable UI allows applications to be developed without being tied to a particular screen size (until S60 2nd Edition, Feature Pack 3, applications were developed for a standard screen size), and this provides some future-proofing for the applications if they are ported to devices of varying dimensions.

The feature discovery APIs allow for querying a device to find which APIs are available. This capability allows a developer to maintain a single code base instead of having to use conditional compilation, which fragments the code base. In effect, these APIs extend the Develop/Optimize paradigm, which developers have been encouraged to adopt, allowing for the integration of future-proofing techniques in basic application design.

## 3. APIs and other changes

S60 3rd Edition supports a number of new C++ APIs and several Java™ Specification Request (JSR) APIs for the Java platform, and it also provides new Open Mobile Alliance (OMA) and Bluetooth technology standards support, allowing developers to create new types of applications and offer additional features in existing applications.

With the addition of Web Run-Time (WRT) to S60 3rd Edition, Feature Pack 2, Web widgets can be run on the S60 platform, opening up new options for delivering Web 2.0 features and services.

### 3.1 C++ APIs in S60 3rd Edition

#### 3.1.1 Location APIs

Location APIs offer developers the option of allowing applications to respond to a consumer's current location and use location-related information. This capability has huge potential for a vast range of services, especially locality-based data queries — for example, finding the nearest restaurant or cinema. The main API for Symbian OS applications is the Location Acquisition API, which has been included since S60 2nd Edition, Feature Pack 2.

S60 3rd Edition introduces the following new location features for developers:

- *Landmarks API* — Via this API, Symbian applications can create, modify, delete, and categorize the landmarks of favorite locations, such as hotels and restaurants.
- *Landmarks Search API* — This API enables Symbian applications to search for landmarks that have been stored on a device by their names, locations, and categories.
- *Basic Location Info Display (BLID) Satellite Info API* — This API provides a UI that displays satellite information in a manner similar to handheld global positioning system (GPS) receivers.
- *Landmarks UI Add, Edit, and Select APIs* — With these APIs, users can add, delete, modify, select, and send landmarks. Adding a landmark allows a user to create one from the current location or manually add the details. The Landmarks UI application also provides APIs for using features from Symbian applications.
- *Landmarks Messaging and Download* — Users can send, receive, and forward landmarks via multimedia messaging service (MMS), e-mail, Bluetooth connections, and infrared (IR), and they can download landmarks from Web pages.

#### 3.1.2 MMF DRM API

The multimedia framework (MMF) DRM API gives third-party music and video players access rights to playback functionalities for content that is protected by digital rights management (DRM).

### **3.1.3 Public SIP API and SDK**

For better interoperability of Session Initiation Protocol (SIP) services, S60 3rd Edition implements a standard public SIP API and SDK for SIP application development.

### **3.1.4 Light API**

An API that controls the turning on and off of the backlight is now available. This removes the need to monitor inactivity time when the user wants to keep the backlight on.

### **3.1.5 Web services APIs**

Web services APIs offer ways to establish connections and to store, retrieve, and manage service information.

### **3.1.6 IM APIs**

Instant messaging (IM) APIs allow developers to use and launch IM via ECom.

### **3.1.7 Exif API**

The Exif API supports Exchangeable Image File (Exif) headers for compressed JPEG files. Exif is intended to enhance interoperability between imaging devices — especially digital cameras — most of which use the Exif format.

### **3.1.8 Find Item API**

The Find Item API enables parsing of phone numbers, e-mail addresses, and URLs from text.

## **3.2 C++ APIs in S60 3rd Edition, Feature Pack 1**

### **3.2.1 Gallery Content Listing API**

The Gallery Content Listing API allows developers to use the new Gallery feature of S60 3rd Edition. The Gallery feature lists all media on a device in a single Gallery browser.

### **3.2.2 Optical Character Recognition API**

The Optical Character Recognition API allows developers to obtain the text elements from images captured with device cameras.

### **3.2.3 Central Repository Notification Handler API**

Access to the Symbian OS Central Repository has been improved and simplified with the Central Repository Notification Handler API.

### **3.2.4 Profiles Engine API**

Direct manipulation of the content of profiles, addition of profiles, and setting of active profiles can be achieved in third-party applications with the Profiles Engine API.

### **3.2.5 Screen Mode API**

The Screen Mode API allows an application to get and set device screen resolution, orientation, and color depth. It is used where an application has not been written using the scalable UI guidelines and APIs, and allows developers to implement applications in one of two ways. The first is to use the Screen Mode API to determine a device's screen characteristics and then programmatically adjust how the UI scales to the display. The alternative is to write a fixed-size UI and then set the device to display this UI size only (with a blank border if the physical screen is larger than the program's fixed UI). This second approach effectively restricts the application UI size to the smallest available S60 screen size, and as a result, this approach is not recommended. The Screen Mode API is of most use to developers who have created custom controls that are not scalable or for situations in which the S60 UI scalability does not produce the desired outcome.

### **3.2.6 OpenGL v1.1 API**

The OpenGL v1.1 API updates OpenGL functionality from v1.0 to v1.1.

### **3.2.7 Browser Control API**

The Browser Control API is updated to provide control of the Web Browser for S60, which is now the default browser for the S60 platform.

### **3.2.8 Telnet Engine API**

The Telnet Engine API allows applications to create Telnet sessions with features that connect to, read from, write to, and disconnect from a Telnet server. The API can also get and set the Telnet configuration.

### **3.2.9 App Framework Animation API**

The App Framework Animation API allows an application to make use of an ECom animation plug-in.

### **3.2.10 OBEX API (updated)**

The OBEX API has been updated to provide additional features for accessing Bluetooth connections, IR, and Universal Serial Bus (USB) transport information, along with support for novel OBEX transport mechanisms.

### **3.2.11 OBEX MTM API (updated)**

The OBEX MTM API has been updated to allow the maintenance of a list of OBEX headers.

### **3.2.12 SIP APIs**

The SIP APIs, which were previously part of the S60 platform, are now included in Symbian OS.

### 3.3 C++ APIs in S60 3rd Edition, Feature Pack 2

#### 3.3.1 RGA APIs

The Real-Time Graphics and Audio (RGA) APIs are a collection of C++ APIs targeted at game developers. In conjunction with Open C (see Section 3.4, “Open C”), these APIs enable developers to work with familiar C/C++ technology to create rich games and other applications with rich graphical or audio content.

The APIs include features for:

- Obtaining device configuration and status information, such as the display configuration and vibration feature.
- Handling keyboard and text input.
- Standard- and high-resolution timers.
- DLL creation and memory management.
- Rendering raster images.
- Playing, recording, and mixing audio.
- Playing and recording video.
- Displaying 2D vector graphics.
- Displaying 3D graphics.

#### 3.3.2 AIW framework APIs

Application Interworking (AIW) allows the embedding of functionality from one application into another. The APIs supporting AIW include:

- AIW Criteria API.
- AIW Generic Parameter API.
- AIW Service Handler API.

Initial support for AIW capabilities is provided by the AIW Dial Service Consumer API and Map and Navigation AIW API.

#### 3.3.3 Map framework APIs

The Map framework APIs enable applications to access map, navigation, and geocoding services via client-side APIs and AIW. The framework consists of the following APIs:

- *Map and Navigation API* — This API enables a range of mapping and navigation operations. These include: show map, show on map, select from map, and go to operations.
- *Map and Navigation AIW API* — This API enables map and navigation information to be included in other applications using the AIW framework.
- *Geocoding API* — This API provides bidirectional coding of strings and addresses to landmarks.

- *Map and Navigation Provider Discovery API* — This API provides a set of enablers for integration of Map and Navigation Provider services.

### 3.3.4 Application network-level roaming APIs

This collection of APIs is designed to allow applications to implement seamless network-level roaming. The APIs provided are:

- *Connection Settings API* — This API provides methods to read connection settings from `CommsDat`.
- *Connection Settings UI API* — This API provides a UI dialog that lets the user select connection settings to be saved in an application's settings.

### 3.3.5 Middle Softkey API

The Middle Softkey API enables use of the new middle softkey.

### 3.3.6 Status Pane API

The Status Pane API enables use of the new status pane.

### 3.3.7 Other updated APIs

Improvements and modifications have also been made in the following APIs:

- DRM Helper API.
- Light API.
- Platform Environment API.
- Connection Manager API (via Symbian OS).

## 3.4 Open C

Open C is an extension of the POSIX libraries for Symbian OS. Open C provides a set of standard POSIX and middleware C libraries from nine well-known C libraries: `libc`, `libdl`, `libpthread`, `libm`, `libz`, `libcrypt`, `libcrypto`, `libglib`, and `libssl`. The implemented functions have been carefully chosen, after extensive analysis of open source projects, to provide a complete set of commonly used functions.

The implementation of Open C allows developers to reuse software assets and thereby increase productivity. Such assets may include a developer's own library of functions and open source code. In addition, because C libraries are commonly used to create applications on other platforms, such as Linux or Microsoft Windows, the Open C support significantly simplifies the process of porting existing applications to the S60 platform.

Open C also reduces the knowledge of Symbian C++ required to create S60 applications. Developers now can create much of their applications' business logic using familiar C library functions, while relying on Symbian and S60 APIs to create the application UIs and to provide access to native Symbian and S60 functions.

Open C was introduced as a native feature in S60 3rd Edition, Feature Pack 2, with a plug-in extension for devices and SDKs based on S60 3rd Edition and S60 3rd Edition, Feature Pack 1.

## 3.5 Java™ APIs in S60 3rd Edition

### 3.5.1 Location API for J2ME™ (JSR-179)

The Location API for J2ME™ (JSR-179) allows for writing location-based applications for the Java platform. The API provides methods for determining the current location of a device and provides access to landmark information.

### 3.5.2 Security and Trust Services API for J2ME™ (JSR-177)

The Security and Trust Services API for J2ME™ (JSR-177) extends the security features of Java™ Platform, Micro Edition (Java™ ME) technology through the addition of cryptographic APIs, digital signature service, and user credential management.

### 3.5.3 SIP API for J2ME™ (JSR-180)

The SIP API for J2ME™ enables SIP networking by allowing Java applications to send and receive SIP messages.

### 3.5.4 Wireless Messaging API 2.0 (JSR-205)

The Wireless Messaging API 2.0 (JSR-205) enhances the Wireless Messaging API (JSR-120) with features such as MMS support. It allows Java developers to create messages that contain images and sounds as well as text.

## 3.6 Java™ APIs in S60 3rd Edition, Feature Pack 1

### 3.6.1 Advanced Multimedia Supplements API (JSR-234)

The Advanced Multimedia Supplements API (JSR-234) provides access to a device's FM radio tuner, as well as music and 3D audio.

### 3.6.2 Scalable Vector Graphics 2D API (JSR-226)

With the Scalable Vector Graphics 2D API (JSR-226), applications can render scalable 2D vector images, including external images in W3C Scalable Vector Graphics (SVG) format. The principal uses for this API are in map visualization, scalable icons, and applications that require scalable and rich animated graphics.

### 3.6.3 Mobile 3D Graphics API for J2ME™ v1.1 (JSR-184)

The updated Mobile 3D Graphics API for J2ME™ v1.1 (JSR-184) allows for rendering 3D graphics at interactive frame rates. It includes facilities for 3D scene management and animation, as well as a file format for efficient over-the-air (OTA) deployment of 3D content.

## 3.7 Java™ APIs in S60 3rd Edition, Feature Pack 2

Improvements in Java™ support in S60 3rd Edition, Feature Pack 2 focus on providing support for the subset of the Mobile Service Architecture (JSR-248). The goal of JSR-248 is to reduce the fragmentation of Java APIs.

### 3.7.1 Mobile Information Device Profile 2.0 (JSR-118)

S60 3rd Edition, Feature Pack 2 provides support for MIDP 2.1, which offers incremental improvements over MIDP 2.0.

### 3.7.2 Mobile Service Architecture (JSR-248), subset

JSR-248 is designed to reduce Java fragmentation by defining a set of Java technologies for high-volume mobile devices. The subset of JSR-248 has been implemented. It encompasses JSR-118, JSR-184, JSR-205, and JSR-226, as well as PDA Optional Packages for the J2ME™ Platform (JSR-75), Java™ APIs for Bluetooth (JSR-82), Mobile Media API (JSR-135), and Connected Limited Device Configuration 1.1 (JSR-139).

### 3.7.3 Scalable 2D Vector Graphics API for J2ME™ (JSR-226)

This updated version of the Scalable 2D Vector Graphics API for J2ME™ (JSR-226) provides support for JSR-248.

### 3.7.4 Java™ APIs for Bluetooth (JSR-82)

Java™ APIs for Bluetooth (JSR-82) has been updated to v1.1. This update adds compliance with JSR-248.

## 3.8 S60 customization framework

S60 3rd Edition provides rich and cost-efficient operator customization. In addition to the existing branding possibilities — for example, themes and the Active Idle screen — there is a new customization framework that provides:

- Look-and-feel customization.
- Software settings.
- Feature variation.
- S60 application extensions.
- Add-on application-development environment.

This package also enables extensive operator branding and service provision as part of the customization process.

## 3.9 Bluetooth standard v1.2 support

S60 3rd Edition implements Bluetooth standard v1.2 support. The main features of v1.2 are:

- Enhanced voice processing that improves the quality of voice connections.
- Adaptive frequency hopping that reduces interference between wireless technologies.
- Enhanced quality of service that increases performance and traffic management scheduling.
- Anonymity mode that increases connection security by masking the physical address of a radio.

### 3.10 OMA standards support

#### 3.10.1 OMA DRM 2.0: base services

OMA DRM 2.0 base services are services of the OMA DRM 2.0 Enabler Release that provide added trust and security for premium content. OMA DRM is a standard for protecting data from unauthorized access and copying. Support for OMA DRM 2.0 is device-dependent.

#### 3.10.2 OMA SyncML DS v1.2

The OMA's SyncML Data Synchronization (DS) standard defines a protocol for synchronization of personal information between mobile devices and personal information manager (PIM) servers. The synchronization client supports OMA DS v1.1.2 and v1.2, which is still under development; this enables better content and filtering support for PIM data.

### 3.11 Interoperability with PCs and Microsoft Exchange Server

S60 3rd Edition introduces support for Microsoft technologies; these technologies will enhance the interaction between mobile devices and PCs. Nokia is providing support for Microsoft ActiveSync, allowing users to synchronize their S60 3rd Edition devices with e-mail, as well as calendar and contact information stored on Microsoft Exchange Server 2003.

### 3.12 Flash Lite from Adobe

S60 3rd Edition, Feature Pack 1 introduced Flash Lite 2.0 from Adobe as an optional component of the S60 platform. As such, Flash Lite 2.0 may not be implemented on all S60 3rd Edition, Feature Pack 1 devices. Any device that is shipped without Flash Lite 2.0 can still make use of the application, once the device's user has installed it.

S60 3rd Edition, Feature Pack 2 improves Flash Lite support with the inclusion of Flash Lite 2.1, again as an option. Flash Lite 2.1 delivers support for inline editing, XML sockets, metadata, and transparency.

A Flash Lite Framework API, which allows developers to interact with the Flash Lite 2.0 player, is present as part of the S60 platform. However, it works only when a device includes the optional Flash Lite 2.0 player, either as an original component or installed by the device's user.

### 3.13 Web widgets

Web widgets are lightweight applications created using standard Web technologies that are used to create Web pages, such as HTML, Cascading Style Sheets (CSS), JavaScript™, and Asynchronous JavaScript™ and XML (Ajax). On an S60 3rd Edition, Feature Pack 2 device, Web widgets are executed using WRT. WRT is based on the open source WebKit — the same technology that powers the Web Browser for S60.

Web widgets are delivered as a ZIP archive with the extension `*.wgz` that S60 device users download and install in the same way they would for any other S60 application. Once installed, Web widgets run like any other S60 application and provide S60 device users with a full Web experience that allows them to personalize the content and services they access.

After optimizing for screen size and widget navigation, Web developers can easily port desktop Web widgets to the S60 platform. Tools for Web widget development will be available from Forum Nokia in the third quarter of 2007.

More information on widgets and WRT can be found on the Forum Nokia Web site's WRT page (<http://www.forum.nokia.com/webruntime>).

### **3.14 API documentation**

The S60 documentation delivered in the SDK was significantly improved in S60 3rd Edition, Feature Pack 1. These changes — both in the structure and content of the documentation — include the following:

- Simplification of the documentation structure for each API, with a single document providing all the relevant information.
- Use of a common template for each API.
- Complete documentation for each API, with no sections omitted.
- A clear, standard, and correct code example for each API.

Overall, these should make it easier for developers to find the information they need, quickly and efficiently.

---

## 4. Conclusion

S60 3rd Edition presents developers with both opportunities and challenges. This paper outlines both. S60 3rd Edition offers developers the potential of a significantly expanded customer base, because S60 devices will be secure and available across a widening range of market segments. At the same time, the introduction of a new ARM binary interface and platform security requires developers to change their development tools and may mean additional certification requirements for some applications.

S60 devices are seeing increased market segmentation, because now the devices are capable of supporting variations in hardware. Single- and dual-chip devices and a wider range of UI specifications allow different device configurations with a range of form factors.

A broader range of segmented devices is bringing new opportunities to developers for all manner of applications. The availability of enhanced security and devices such as the Nokia Eseries devices will also encourage more business use. The S60 platform has excellent music and video capabilities that will allow developers to maximize the entertainment value of their applications.

## 5. Terms and abbreviations

Term or abbreviation	Meaning
ABI	Application Binary Interface
ARM <sup>®</sup>	Advanced RISC Machines
DRM	Digital rights management
EKA	EPOC Kernel Architecture
Exif	Exchangeable Image File
JSR	Java™ Specification Request
MMF	Multimedia framework
OMA	Open Mobile Alliance
RVCT	RealView <sup>®</sup> compilation tools
SIP	Session Initiation Protocol
SyncML	Synchronization Markup Language

---

## 6. Evaluate this resource

Please spare a moment to help us improve documentation quality and recognize the resources you find most valuable, by [rating this resource](#).