S60 3rd Edition: What's New for Developers

Version 1.2
December 16, 2005

S60 platform

Legal Notice

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. Furthermore, information provided in this document is preliminary and may be changed substantially prior to final release. 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

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

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 primarily at developers who are about to create applications for S60 3rd Edition or who want to port an existing application to the new version of the platform.

2. General Developer Issues

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

2.1 ABI Compiler for ARM® Architecture

From S60 3rd Edition onward, a compiler based on the Application Binary Interface (ABI) for the ARM® Architecture will be used to compile 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 is optimized to meet the unique needs of embedded application development.

In adopting the ARM ABI compiler 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 amongst 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 v9.1, which features a new real-time kernel, 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 new EKA2 kernel has also resulted in improved emulation. The Win32 emulator for EKA2 shares the same core kernel code, ensuring a more faithful emulation of the target hardware; in particular, process emulation is supported. This enhancement will mean a reduction in development time for some projects, because less on-target debugging is required. It also means that there should be a reduced need to write WINS platform-specific code.

The EKA2 kernel will also enable S60 devices to be built on a single-processor architecture, potentially reducing the bill of materials for a S60 device. A reduction in device costs will enable S60 manufacturers to create phones in the mid range, which brings the potential for significantly higher volumes of devices to be shipped. 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 end users 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 will have, with the right verification, access to a greater range of APIs. The changes also provide some protection from unwanted side effects caused by defects in applications. From a business point of view, the enhancements increase the reputation of the S60 platform as a stable, secure operating

system. 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 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 are 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 trusted applications are able to use certain APIs and system resources. The user (that is, the person who installs the application on the phone) 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 capabilities will be available only to Symbian Signed applications.

There are four sets of capabilities.

- **Open.** These capabilities are open to all applications; applications do not need to be Symbian Signed.
- **Basic.** These capabilities (with the exception of full location information) can be granted by the user to the application on installation and are known as unsigned sandbox. Symbian Signed applications can access all these capabilities without requiring the user to grant permission.
- **Extended.** These capabilities require an application to be Symbian Signed before they can be accessed.
- **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, www.forum.nokia.com/documents.

2.4 Tools

As a result of the new ABI-based compiler for the S60 3rd Edition, developers will need to obtain new compiler tools that conform to this standard. Developers will be able to choose from tools conforming to the ARM C/C++ ABI standard.

The first to support this standard is the ARM RealView® 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 will be made available for free with the S60 3rd Edition SDK. Borland Software Corporation and Microsoft Corporation are also expected to produce tools compliant with this compiler standard.

The new Carbide.c++ tools from Nokia, due for release in 2006, will also provide support for ARM C/C++ ABI standard compliers.

2.5 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.

2.5.1 Certification procedures

There are two main sources of additional costs for developers wishing to create applications for S60 3rd Edition: the new tools that will be required for development (see Section 2.4, "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 listed below.

- Emulator development — This stage incurs no extra costs for developers, beyond those associated with tools changes.
- Device testing — To enable testing on a device, developers must obtain a Symbian Developer Certificate. 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 — Granting Symbian Signed status to an application will be undertaken by a test house for a fee. The cost per test cycle should be about $255 (210.10 euros) for an application and $33 (27.19 euros) for content.

Symbian Signed is recommended for all applications, but is not necessary for applications using APIs only from the Open and Basic (unsigned sandbox) capability sets. However, Symbian Signed removes the user installation warning and the need to grant permission to use Basic capabilities, making it more likely the user will install the application.

2.5.2 Developer benefits

While creating applications for S60 3rd Edition may incur additional costs, for most developers, the benefits of the changes in the new version of the platform will outweigh them.

- 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 developers of enterprise and other data-sensitive applications.
- Application efficiency — With its real-time capabilities, the new 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 open to developing parties. There are many APIs that will now be publicly accessible so that the functionality can be directly accessed. This will allow enhancements to all types of applications.
- Increased market confidence — Certification, effectively a requirement for complex applications, will promote increased consumer and distributor confidence that all such applications will work efficiently, safely, and correctly. As the market is still in its initial growth phase, issues of trust and reliability are becoming increasingly 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 now will promote its established reputation as a key innovator in the market. Clearly, it is to the advantage of development companies to take advantage of this situation as soon as possible.

2.6 Code Base Maintenance and Future-Proofing

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

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 application should it be ported to devices of varying dimensions.

The Service Discovery APIs allow for querying the device to find which APIs are available. This capability allows developers 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. API 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 provides new Open Mobile Alliance (OMA) and Bluetooth technology standards support, which will allow developers to create new types of applications and offer additional features in existing applications.

3.1 C++ APIs

3.1.1 Location APIs

Location APIs offer developers the option of allowing applications to respond to a user's current location and use location-related information. This 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 in S60 2nd Edition, Feature Pack 2 and later.

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

- Landmarks API — Via this API, Symbian OS applications can create, modify, delete, and categorize the landmarks of favorite locations — for example, hotels and restaurants.
- Landmarks Search API — This API enables Symbian OS applications to search for landmarks that have been stored on the device by their name, location, and category.
- Basic Location Info Display — This application provides basic location-information displays for the end user and contains features similar to handheld Global Positioning System (GPS) receivers.
- Landmarks UI — With this application, users can add, delete, modify, and send landmarks. Adding a landmark allows the 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 OS applications.
- Landmarks Messaging and Download — Users can send, receive, and forward landmarks over Multimedia Messaging Service (MMS), e-mail, Bluetooth, and IrDA, and they can download landmarks from Web pages.

3.1.2 MMF DRM API

The Multimedia Framework (MMF) API gives third-party music and video players access rights to playback functionalities for content 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 Backlight APIs

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

This API supports Exchangeable Image File (EXIF) headers for compressed JPEG files intended to enhance interoperability between imaging devices — especially digital cameras — many of which use this 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 Java APIs

3.2.1 Location API for JME™ (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.

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

The Security and Trust Services API (JSR-177) extends the security features of Java™ 2 Platform, Micro Edition (JME™) through the addition of cryptographic APIs, digital signature service, and user credential management.

3.2.3 SIP API for JME™ (JSR-180)

The SIP API for JME™ (JSR-180) enables SIP networking by allowing Java applications to send and receive SIP messages.

3.2.4 Wireless Messaging API 2.0 (JSR-205)

The Wireless Messaging API (WMA) 2.0 (JSR-205) provides access to MMS. This allows Java developers to create messages that contain images and sounds as well as text.

3.2.5 Scalable 2D Vector Graphics API for JME™ (JSR-226)

The Scalable 2D Vector Graphics API for JME™ (JSR-226) is concerned with the rendering of scalable 2D vector graphics, including image files in the Scalable

Vector Graphics (SVG) format of the World Wide Web Consortium (the W3C). Areas of application include map visualization, scalable icons, and other advanced graphics manipulation.

3.2.6 Mobile Service Architecture for CLDC (JSR-248)

The Mobile Service Architecture for CLDC (JSR-248) creates a mobile service architecture and platform definition for the high-volume wireless devices continuing the work started in JSR-185, enhancing the definition with new technologies.

3.3 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.

3.4 Bluetooth Standard V1.2 Support

S60 3rd Edition implements Bluetooth standard v1.2. The main features of this specification 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.

3.5 OMA Standards Support

3.5.1 OMA DRM version 2.0: base services

OMA DRM version 2.0 base services are services of the OMA DRM version 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.

3.5.2 OMA SyncML DS V1.2

A standard of the OMA, SyncML Data Synchronization (DS) 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 the 1.2 version that is still under development; this will enable better content and filtering support for PIM data.

3.5.3 SyncML API

The SyncML API supports OMA DS V1.2. With this API, developers will be able to call on functions to carry out synchronization tasks from inside their applications. SyncML DS features for this release are:

- Backward compatibility with SyncML DS V1.0.1.
- Compatibility with extensions for SyncML e-mail.
- Compatibility with current and future servers such as IBM WebSphere and Oracle Collaboration Suite SyncML server.

3.6 Interoperability with Microsoft PCs and Exchange Server

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

4. The Future

In October 2005 Nokia announced S60 3rd Edition, Feature Pack 1. It includes a Web browser based on open source components, as well as device management extensions, and a flexible XML-based UI customization framework. Primarily aimed at supporting the migration of the S60 platform to the mid range, it also offers new features to developers.

In addition, during 2006 the features of the software platform on the Nokia 7710 widescreen smartphone (formerly known as the Series 90 Platform) will become available as part of the S60 platform.

5. Conclusion

The introduction of 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 wider range of market segments. At the same time, the introduction of a new ARM binary and platform security will require developers to change their development tools and may impose additional certification requirements on some applications.

S60 devices are likely to see increased market segmentation because devices are now 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 will bring new opportunities to developers of 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.

6. Terms and Abbreviations

Term or abbreviation	Meaning
ABI	Application Binary Interface
ARM	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® Compilation Tools
SIP	Session Initiation Protocol
SyncML	Synchronization Markup Language