CHAPTER 3: SOFTWARE AND HARDWARE REQUIREMENT
Personal Profile (PP) is part of Java ME and supports resource-constrained devices with a graphical user interface toolkit based on AWT. When combined with the Connected Device Configuration (CDC), Personal Profile provides a complete Java ME application environment for consumer products and embedded devices.

It has more powerful UI resources to develop applications than MIDP.
Java ME is based on different APIs (JSR) available in mobile devices, apart from the configuration (like CDC or CLDC) and the profile (like MIDP or Personal Profile).

You have to be sure if a device has installed an API before using it. You can check over platform information, for example for Series 40 or S60, to find which JSRs are available.

There are two standards defined to reduce fragmentation, grouping many of the APIs in one platform name:

- Java Technology for the Wireless Industry (JTWI)
- Mobile Service Architecture (MSA)

Many APIs are specific to some vendor or operator. In this category, we can find:

- Nokia UI API

There are 82 JSRs defined in JCP about Java ME. Many of them are still in draft and will be available in future devices and some were deprecated.

The most common additional APIs found on mobile devices are:

- Wireless Messaging API (WMA) (JSR 120)
- Mobile Media API (MMAPI) (JSR 135)
- FileConnection & PIM API (JSR 75), also known as PDA Optional Packages
- Bluetooth API (JSR 82)
- Security and Trust Services API (SATSA) (JSR 177)
Java ME is based on different APIs (JSR) available in mobile devices, apart from the configuration (like CDC or CLDC) and the profile (like MIDP or Personal Profile).

You have to be sure if a device has installed an API before using it. You can check over platform information, for example for Series 40 or S60, to find which JSRs are available.

There are two standards defined to reduce fragmentation, grouping many of the APIs in one platform name:

- Java Technology for the Wireless Industry (JTWI)
- Mobile Service Architecture (MSA)

Many APIs are specific to some vendor or operator. In this category, we can find:

- Nokia UI API

There are 82 JSRs defined in JCP about Java ME. Many of them are still in draft and will be available in future devices and some were deprecated.

The most common additional APIs found on mobile devices are:

- Wireless Messaging API (WMA) (JSR 120)
- Mobile Media API (MMAPI) (JSR 135)
- FileConnection & PIM API (JSR 75), also known as PDA Optional Packages
- Bluetooth API (JSR 82)
- Security and Trust Services API (SATSA) (JSR 177)
Java Technology for the Wireless Industry (JTWI) is a JCP specification (JSR 185) that defines what APIs must include a Java ME device platform after year 2003.

A JTWI device must include:

- MIDP 2.0
- WMA 1.1 (JSR 120)
- CLDC 1.0 (JSR 30)

Optional:

- MMAPI 1.1 (JSR 135)
- CLDC 1.1 (JSR-139)

It recommends some minimum requirements on device:

- 125x125 screen resolution
- 4096 colors (12 bits)
A more recent evolution of this standard is Mobile Service Architecture.

Devices that implement JSR 185 should allow MIDlet JARs up to 64 KB, with application descriptors (JADs) up to 5 KB, and each MIDlet suite should be allowed to use up to 30 KB of persistent storage (RMS).

Mobile Service Architecture (MSA) is an specification from JCP to reduce Java ME device fragmentation and allows vendors to distribute new devices under this umbrella. So, if a device is a MSA phone, you are sure what additional APIs have.

Like its predecessor, Java Technology for the Wireless Industry, MSA is an umbrella over a collection of familiar, updated, and new JSRs that cooperate to support applications with a wide range of standardized capabilities in Java ME.

There are two MSA standards, depending on the platform.
Figure 3.1. Java ME mindmap

The mindmap shows the relationship between CLDC, MIDP and the different Java Optional Packages specified in the JSRs.
The following tools may be part of the extended tool chain for mobile application development in different platforms:

for Symbian OS C++ Development

- CodeWarrior for Symbian OS
- Carbide
  - Carbide.c++ for Symbian OS
    - Carbide.c++ product chart
    - Carbide.c++ FAQ
    - UI Designer
    - Performance Investigator
  - Carbide.vs (formerly Nokia Developer Suite for Symbian OS)
- Microsoft
  - Visual Studio 2003
  - Visual Studio 2005
  - Visual C++ Express
- Symbian C++ Development on Linux
For Themes development

- Carbide.ui - supporting both S60 and Series 40

For Maemo Development

- Laika plugin for Eclipse

For Java ME Development

- Carbide
  Carbide.j (formerly Nokia Developer Suite for Java)
- NetBeans with Mobility Pack
- EclipseME, a plugin for Eclipse
- JBuilder from CodeGear (formerly Borland)
- JDeveloper from Oracle
- Mobile Interaction Suite from Extransit
- InfoNU from IntelliTech Software

For Flash Lite Development

- Flash Professional (version 8 or CS3)
- Flash MX 2004 (requires updates)

For WAP Development

- Adobe Dreamweaver CS3 with Device Central CS3
- Microsoft
  - Visual Studio 2005
  - Visual Web Developer Express
For Python / PyS60 Development

- Eclipse + PyDev

- GCC
- GCCE
- Microsoft C++ compiler
- Nokia x86 compiler
- RVCT

- Abld.bat
- Bldmake
- CreateSIS
- Devices.exe
- Elftran
- MakeKeys
- MakeSIS
- Petran
- SignSIS
- UnSIS
- 中文 SignSIS

- Nokia Connectivity Framework
- Nokia Mobile Internet Toolkit
- Command prompt (cmd.exe)

- makefile
- ActivePerl
- Java Runtime Environment (JRE)
Nokia Series 40 is a platform for mobile phones based on the same core, functionality and some technical details. It is found in more than 100 million devices. There was a Series 30 that doesn't exist anymore. Some of its devices (as Nokia 3595) were upgraded to Series 40.

Series 40 on the surface appears to be a simpler operating system than the higher end S60 and Series 80. Series 40 devices differ from the Symbian-based platforms in that they do not support true multi-tasking and do not have a native code API for third parties and thus do not support installable applications other than MIDlets that are written in Java or applications in Flash Lite. It is primarily because of the former that Series 40 appears to be more responsive and faster than other Nokia platforms.

The Series 40 platform offers a mass-market opportunity for Java™ developers, developers of applications for Flash Lite from Adobe, and content creators. Devices based on the Series 40 platform accounted for a large portion of the more than 540-million platform devices Nokia had cumulatively shipped by the end of 2006.

Java developers have access to MIDP and CLDC technology, with an array of JSRs that provide additional communication, messaging, media, and graphics capabilities.

Media developers can deliver Web, messaging, and Flash Lite content, as well as streaming video and audio, using the latest industry standards. The user interface of Series 40 devices can also be personalized with themes, animated screen savers, and wallpaper.

All this is supported by OMA DRM to protect developers' intellectual property.

Series 40s generation is as follow:

- 1 Series 40 1st Edition
- 2 Series 40 2nd Edition
3 Series 40 3rd Edition
   o 3.1 Series 40 3rd Edition Feature Pack 1
   o 3.2 Series 40 3rd Edition Feature Pack 2
4 Series 40 5th Edition

The first edition of this platform supports:

- Java ME with MIDP 1.0, CLDC 1.0 and Nokia UI API.
- WAP Browsing with WML support.
- The classic screen resolution is 128x128

Its devices support:

- Java ME with MIDP 2.0, CLDC 1.0, Wireless Messaging API (WMA), Mobile Media API (MMAPI) and Java APIs for Bluetooth (JSR 82) (where the device supports the technology).
- WAP Browsing with WML and XHTML MP (Mobile Profile) support.
- The classic screen resolution is 128x160

Its devices support:

- Java ME with MIDP 2.0, CLDC 1.1, Wireless Messaging API (WMA), Mobile Media API (MMAPI), Java APIs for Bluetooth (JSR 82), FileConnection and PIM APIs and Mobile 3D Graphics API (JSR 184).
- WAP 2.0 Browsing with WML, XHTML MP (Mobile Profile) and ECMAScript support.
- Screen resolutions can be 128x160, 208x208, 240x320.
Series 40 3rd Edition Feature Pack 1

Extending the 3rd Edition, this pack adds:

- For Java ME, supports for Wireless Messaging API 2.0 (JSR 205) and Scalable 2D Vector Graphics API (JSR 226).
- Flash Lite applications and content supporting version 1.1 of the Adobe's platform
- Support for Themes creation

Series 40 3rd Edition Feature Pack 2

- Java ME adds support for JSR 177 (APDU package only)
- Flash Lite 2.0 support

Series 40 5th Edition extends Nokia's platform promise, with the introduction of MIDP 2.1 and the subset of the Mobile Service Architecture (JSR-248). JSR-248 provides a common implementation of seven popular JSRs. These implementations are shared with S60 3rd Edition, Feature Pack 2, and it is expected that other manufacturers of Java™ phones will implement JSR-248. This means that developers will be able to create applications for Nokia platforms that will also run on other manufacturers' devices, with little or no change.

In addition, Series 40 5th Edition delivers the Advanced Multimedia Supplements (JSR-234) and enhanced versions of the Java™ APIs for Bluetooth (JSR-82), the Mobile Media API (JSR-135), the J2ME™ Web Services Specification (JSR-172), and the Security and Trust Services API for J2ME™ (JSR-177). Series 40 5th Edition, Feature Pack 1 also adds the Content Handler API (JSR-211).

In addition to Java technology enhancements, Series 40 5th Edition provides other opportunities to developers of applications for Flash Lite, with the introduction of the Flash Lite 2.1 Player from Adobe. In addition to playing stand-alone Flash Lite content and
applications, and offering animated screen savers, Flash Lite can also be used to create animated "organic" wallpaper for Series 40 5th Edition devices.

Series 40 5th Edition will allow developers to create a wider range of applications than ever before. Series 40 devices have always offered an advanced mobile experience to consumers; now features such as JSR-172 and JSR-177 will allow Java developers to deliver advanced applications to enterprise users, too. Because of the platform's continued support for 240 x 320-pixel screens, media developers will benefit from the improved viewing experience and realize faster growth of the consumption of content and media. The platform approach, with its uniform implementation of technologies and supporting tools, documentation, and examples, will continue to offer developers the easiest route to seize these market opportunities.

- Java ME adds support for JSR 177 (APDU and CRYPTO), JSR 234 (Music and 3D Audio only)
- Flash Lite 2.1 support

Java™ technology provides developers with a robust cross-platform development tool capable of addressing the most demanding enterprise and consumer application requirements.

With the latest version of the Series 40 platform, developers have access to the following Java Technology:

- PDA Optional Packages for the J2ME™ Platform (JSR-75), with support for the FC and PIM APIs.
- Java™ APIs for Bluetooth v1.1 (JSR-82).
- Wireless Messaging API (JSR-120).
- An updated Mobile Media API (JSR-135).
- J2ME™ Web Services Specification (JSR-172).
- Security and Trust Services API for J2ME™ (JSR-177), including the SATSA-APDU and SATSA-CRYPTO optional packages.
- Mobile 3D Graphics API for J2ME™ (JSR-184).
- Java™ Technology for the Wireless Industry (JSR-185).
Wireless Messaging API (JSR-205).
- Content Handler API (JSR-211).
- Scalable 2D Vector Graphics API for J2ME™ (JSR-226).
- Advanced Multimedia Supplements (JSR-234), providing 3D audio and music support.

The Java implementation on the Series 40 platform shares many common features with the implementation on the S60 platform and the Series 80 platform. This allows developers to easily extend the reach of their applications to the entire platform range.

Flash Lite from Adobe allows developers to deliver rich multimedia applications and content to Series 40 devices.

With support for Flash Lite 2.1, developers can create rich stand-alone applications and content, as well as animated screen savers and animated "organic" wallpaper. Using fscommand2, Flash Lite content can access and control various phone parameters. Flash Lite applications can also initiate various actions, such as making a phone call, allowing applications to take advantage of Series 40 devices' mobile connectivity.

The Series 40 platform provides media developers with options to create browsing and messaging applications, deliver streaming audio and video, and create themes to personalize devices.

For browsing applications, developers can take advantage of support for OMA Browsing v2.1, providing WAP 2.0, XHTML-MP, and HTML 4.01 browsing with wTCP/IP. ECMAScript support is also available, as is support for cookies.

For messaging applications, developers have OMA MMS v1.2 compliant messaging with SMIL to create messages as animated slide shows.

For content, developers can deliver streaming audio in AMR-WB and AAC formats, and video in H.263 format.
For themes, developers can customize the look and sound of the user interface, and add animated screen savers and animated "organic" wallpaper.

**Screen resolutions**

From Series 40 3rd Edition, Feature Pack 2 onward, two UI screen resolutions have been used on Series 40 devices, as illustrated here:

**128 x 160 pixels**

**Figure 3.2.** This enlarged version of the "classic" Series 40 UI resolution offers extra screen size. The first implementation was on the Nokia 6650 phone.

**240 x 320 pixels**

**Figure 3.3.** This is a large QVGA Series 40 UI resolution. The first implementation was on the Nokia 6265 phone, Nokia 6270 phone, and Nokia 6280 phone.
Series 60 Platform

Overview

The S60 platform is a purpose-built platform for smartphones on Symbian OS. It is currently one of the leading mobile platforms in the world. The S60 platform is a complete smartphone reference design, developed by Nokia Mobile Software, and is currently being licensed by several of the world's key mobile device manufacturers.

The S60 platform guarantees to developers that specific elements will be present in every device based on a particular platform edition and feature pack. It does this using the architecture shown in Figure 3.4, which consists of the S60 UI style, S60 Applications, S60 Application Services, S60 Java™ Technology Services, S60 Platform Services, and Symbian OS Extensions built on top of Symbian OS.

![Figure 3.4. schematic diagram of the S60 platform architecture is shown.](image)

The S60 platform brings end users the best smartphone experience available. The platform is implemented in a diverse range of devices and provides application and media developers with a consistent set of technologies. Equally at home delivering advanced enterprise applications, games, or music, the S60 platform gives developers unparalleled opportunities in the mobile space.
Developers can work in C++ (using native Symbian OS APIs and a subset of the POSIX standard libraries provided by Open C), the Java™ language (using MIDP 2.0 with an extensive range of additional JSRs), Flash Lite from Adobe, and Python.

Once applications and content have been created, support for OMA DRM protects developers' intellectual property.

S60 licensees had shipped more than 100 million S60 devices cumulatively by April 2007. As for the overall smartphone market, it is expected to continue growing rapidly; Nokia expects cumulative smartphone shipments to exceed 250 million units in 2008. According to company projections, S60 devices will account for the bulk of this volume.

The S60 platform enables developers to create high-value applications and content with lucrative revenue opportunities. The platform approach — with uniform implementation of technologies and supporting tools, documentation, and examples — requires less effort by developers to grasp these market opportunities.

The S60 platform is a purpose-built platform for smartphones. It supports a large color screen and an intuitive interface, and it incorporates leading-edge communications and device technologies that interoperate safely and respond quickly. Most S60 devices feature:

- At minimum, a 176 x 208-pixel color screen.
- Innovative form design and keypad layout.
- Personal Information Manager (PIM) applications (such as contacts and calendar applications).
- Advanced telephony features.
- Messaging.
- Internet browsing.

Many S60 devices also feature:

- Flash Lite from Adobe player.
- Nokia Push to Talk over Cellular (PoC).
- Digital camera.
- Music player.
Media gallery.
Video recorder.
Sound recorder.
FM radio.
Over-the-air (OTA) synchronization.
Viewer and editor for Microsoft Office documents.

The S60 platform guarantees to developers that specific elements will be present in every device based on a particular platform edition and feature pack. It does this using the architecture, which consists of the S60 UI style, S60 Applications, S60 Application Services, S60 Java™ Technology Services, S60 Platform Services, and Symbian OS Extensions built on top of Symbian OS.

- High resolution screen (>176x208) with high color depth (>16bit)
- Advanced smartphone features, that makes s60 more similar to a computer
- Integrated device manager applications (PIM Personal Information Manager, Phone Browser, etc.) to start immediately to work
- Powerful Multi-Media tools
- Basic phone features enhanced to a new dimension

Symbian OS is a purpose-built smartphone operating system. It has an impressive range of technology features — even in its earliest versions. Its memory-management and multitasking features allow for safe and efficient operation under conditions of constrained resources that typify mobile devices. Different S60 editions have used different versions of Symbian OS; Note that not necessarily all features available in a specific version of Symbian OS are implemented in a particular version of the S60 platform: It is always better to follow the S60 guidelines regarding what is available, rather than to assume that a particular Symbian OS feature has been implemented.
Symbian OS Extensions

The Symbian OS Extensions are a set of capabilities that allow the S60 platform to interact with device hardware functions such as vibration alert, device lights, and battery charge status.

The S60 Platform Services are the fundamental services offered by the S60 platform. These include the:

- **Application Framework Services** — providing the basic capabilities for launching applications and servers, state-persistence management, and UI components.
- **UI Framework Services** — providing the concrete look and feel for UI components and handling UI events.
- **Graphics Services** — providing capabilities for the creation of graphics and their drawing to the screen.
- **Location Services** — allowing the S60 platform to be aware of a device’s location.
- **Web-Based Services** — providing services to establish connections and interact with Web-based functionality, including browsing, file download, and messaging.
- **Multimedia Services** — providing the capabilities to play audio and video, as well as support for streaming and speech recognition.
- **Communication Services** — providing support for local and wide area communications, ranging from Bluetooth technology to voice calls.

The S60 Application Services are a set of capabilities that provide certain basic functionality for S60 applications. These services are used by the embedded S60 Applications and also are available for use in third-party applications. They include:
**PIM Application Services** — providing the fundamental features of PIM applications, including contacts, calendar, and task management, as well as associated functions such as notepad and clock capabilities.

**Messaging Application Services** — providing support for various messaging types, such as short message service (SMS), multimedia messaging service (MMS), e-mail, BIO messages (smart messaging), and instant messaging (IM).

**Browser Application Services** — providing the capabilities to view Web content, including support for Flash Lite from Adobe, video rendering, Scalable Vector Graphics–Tiny (SVG-T) rendering, and audio rendering.

The **S60 Java™ Technology Services** provide support for the Java™ Platform, Micro Edition (Java™ ME) Java™ Technology for the Wireless Industry (JTWI) specification (JSR-185). The S60 platform support includes the Connected Limited Device Configuration (CLDC) 1.1 (JSR-139) specification, and the Mobile Information Device Profile (MIDP) 2.0 (JSR-118) extension to this specification.

Further, the S60 Java Technology Services support a range of additional APIs to enable access to the S60 file system, access to PIM data, use of Bluetooth technology, messaging, audio, video, Web services, security and trust services, location information, Session Initiation Protocol (SIP), and 3D graphics.

The **S60 Applications** are applications embedded within the platform that are available to a device’s user, including PIM, messaging, media applications, and profiles.

This section describes the key technologies implemented in the S60 platform that are available on most S60 devices.
Most S60 devices support these communication technologies:

- Telephony.
- Infrared (IR).
- Bluetooth wireless technology.
- In addition, many S60 devices also support WLAN.

Most S60 devices support these messaging technologies:

- SMS.
- MMS.
- E-mail.
- IM. (The type of support for this varies from device to device.)

All S60 devices support browsing. Early S60 editions supported WAP browsing, while more-recent editions support full HTTP browsing. The Nokia Web Browser, based on open source components from Apple Inc.'s WebCore — the software used in Apple’s Safari browser — provides browsing capabilities in S60 3rd Edition and later.

- Carbide.C++ 1.2 beta
- MS Visual Studio .NET 2003 + Carbide.vs 2.01
- Metrowerks Codewarrior 3.1
- MS Visual Studio 6.0
- Borland C++BuilderX
- Command line tools
Symbian OS was built using C++, and this development language provides the fullest access to the feature-rich S60 platform. All versions of the S60 platform support C++ development, and various integrated development environments (IDEs) are available for each version.

C++ development for the S60 platform requires two components: a suitable development environment and an SDK for the edition and feature pack(s) at which the application will be targeted. SDKs are provided free of charge.

C++ development for the S60 platform is supported by Carbide.C++, a member of the Carbide family of tools from Nokia. Carbide.C++ is based on the Eclipse IDE and is being made available in four versions: Express Developer, Professional, and OEM.

Carbide.C++ Express, a free version, is designed for general-purpose development of Symbian OS applications.

Carbide.C++ Developer is designed for application developers with more stringent application-quality requirements. It provides the ability to perform on device debugging and includes a UI design tool that helps simplify S60 development.

Carbide.C++ Professional is designed for developers who require optimal performance from their applications. The tool includes a performance investigator and crash debugger.

Carbide.C++ OEM is designed for device manufacturers and developers who create fundamental components for device manufacturers. Carbide.C++ OEM had made and available from 2007.

Carbide.C++ Developer, Professional, and OEM are commercial tools.

In addition, Nokia provides Carbide.vs for C++ developers. This tool integrates Symbian OS application development into Microsoft Visual Studio .NET 2003. Carbide.vs is available as a free download.

Nokia has been at the forefront of extending the available Java APIs from within the MIDP development environment.

As with C++ development, Java development for the S60 platform requires a suitable development environment and a suitable SDK. Nokia supports Java development on the S60 platform with Carbide.j, which is free, as are the various SDKs.
Content development options include developing Web pages that are mobile browser friendly and providing items such as themes, wallpaper, and ring tones.

There is already a thriving market for such items, which can provide developers with a relatively quick return on their product-development investments.

The Carbide.UI S60 Theme Edition for Symbian OS, available free of charge, supports theme development. It allows UI designers and artists to create themes for S60 devices.

Other development tools that can be used on the S60 platform include Flash Lite from Adobe, Python, Microsoft Visual Basic, and Microsoft C#.

S60 is the reference platform for Flash Lite, and developers can create Flash Lite 1.0, 1.1, and 2.0 content for various S60 editions and feature packs.

The S60 platform specification includes a UI style and UI libraries (Avkon), thereby promoting a consistent look and feel; however, the specification does not mandate a particular UI. In attempting to future-proof applications, developers should assume that a range of devices of varying UI size, color depth, and input methods will be available, and they should plan and program with scalability in mind. Developers can also create custom components — normally derived from the Avkon libraries.

- Various resolutions, sizes, color depths and shapes (Avkon libraries)

- S60 1st Edition (Version 0.9) - Symbian 6.1
- S60 1st Edition, Feature Pack 1 (Version 1.2) - Symbian 6.1
S60 1st Edition
S60 1st Edition is based on Symbian OS v6.1 and provides this extensive range of technologies:

- PIM applications such as Calendar, Phonebook, Photo Album, To-Do list, and File Manager.
- PC connectivity software:
  - Pinboard.
  - RealPlayer.
  - XHTML Mobile Profile (XHTML MP).

S60 2nd Edition
S60 2nd Edition is based on Symbian OS v7.0s. Its enhancements include:

- Multi homing.
- Java MIDP 2.0.
- Dual IP stacks supporting both IPv4 and IPv6 formats.
- ECom plug-in framework.
- EDGE telephony interface.
- Lightweight multithreaded multimedia framework.
- Support for wideband CDMA (WCDMA).
- WAP 2.0.

Some of the most popular enhancements included in the lead software are:

Multimedia applications — Camera, Image Viewer, RealPlayer, Media Gallery applications, and a voice recorder.
**Wallet** — an application for storing protected personal information. Virtual cards, such as credit cards, can be stored and then used for payment transactions over the Internet.

**Themes** — provide enhancement of personalization via UIs that include themed wallpaper, icons, and bitmaps. These can be applied to give a consistent yet individual look to a device. There have been three feature pack releases for S60 2nd Edition, all of which provide additional lead software.

<table>
<thead>
<tr>
<th>Feature Pack</th>
<th>Features</th>
<th>Example devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature Pack 1</td>
<td>(Symbian OS v7.0s)</td>
<td>Megapixel camera with 4x zoom</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Recording</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Playback of video clips</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nokia 7610 and Nokia 6670 imaging phones</td>
</tr>
<tr>
<td>Feature Pack 2</td>
<td>(Symbian OS v8.0a)</td>
<td>1.3-megapixel camera with 6x zoom,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WCDMA/EDGE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IPv6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nokia 6630 smartphone</td>
</tr>
<tr>
<td>Feature Pack 3</td>
<td>(Symbian OS v8.1a)</td>
<td>Scalable UI support (variable screen sizes)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nokia N90 device</td>
</tr>
</tbody>
</table>
S60 3rd Edition

As the mobile market expands and customers' expectations for better performance from their devices continue to rise, it is clear that reliable large-screen phones with short response times and appropriate levels of data security will become the dominant players.

With each new edition, the S60 platform introduces significant enhancements in all areas, and S60 3rd Edition is no exception. The combination of a real-time kernel and support for important features, such as scalable UIs and location-based services (LBS), brings real benefits in terms of performance and flexibility allied with opportunities for significant device differentiation and, hence, market segmentation.

S60 3rd Edition introduces significant changes in the S60 platform: a new kernel, a new binary, and enhanced security.

New kernel — S60 3rd Edition is based on Symbian OS v9.x, which features a new real-time kernel called EPOC Kernel Architecture 2 (EKA2); EPOC was the original name for Symbian OS. The new kernel allows device manufacturers to create devices with a single-chip architecture, reduce their bills of materials, and offer S60 devices to the mid-market.

New binary — Symbian OS v9.x is based on a new binary: the Application Binary Interface (ABI) standard for the ARM® Architecture. This binary offers significant performance improvements for applications running on S60 devices. This change has created a binary break with all previous versions of Symbian OS and the S60 platform. As a result, applications for earlier versions of the S60 platform will require recompiling, at a minimum, before they can run on S60 3rd Edition.

Enhanced security — S60 3rd Edition introduces Symbian OS platform security, which provides data caging (a secure folder for an application's data) and capabilities defining a number of APIs that require an application to be certified before it can access those APIs. These features offer protection against malicious software. There is also a range of Open Mobile Alliance (OMA) digital rights management (DRM) version 2.0 features provided.

There have been two feature pack releases that provide additional features to S60 3rd Edition.
Feature pack Features Example devices

Feature Pack 1
(Symbian OS v9.2)
- Nokia Web Browser fully integrated
- Flash Lite 2.0 (optional)
- Enhanced UI
- Applications
- Nokia N95 (Multimedia computer)

Feature Pack 2
(Symbian OS v9.x)
- Multiple data stores for Contacts
- Middle softkey
- Unified message editor
- New media player
- Flash Lite 2.1 (optional)
- Map framework APIs
- APIs for seamless connection transition

For C++ developers, S60 3rd Edition also brings support for a set of standard POSIX and middleware C libraries through Open C. This replaces the limited support for POSIX standard libraries provided by Symbian OS (through estdlib).

Open C support is integrated into S60 3rd Edition, Feature Pack 2 and will be available for S60 3rd Edition and S60 3rd Edition, Feature Pack 1 as an add-on.

This feature will enable developers to port components written in standard C to S60 3rd Edition, as well as give developers who are familiar with standard C functions a more straightforward route to development for the S60 platform.

- 1st & 2nd edition compatible with *.sis applications
- 3rd edition compatible with *.sisx (and certified *.sis) applications
More than a dozen device models based on S60 3rd Edition are shipping. Based on state-of-the-art Symbian OS v9, S60 3rd Edition is packed with features and capabilities that let developers reach new markets, bring better performance to their applications, and improve application integrity.

Symbian OS platform security was introduced in S60 3rd Edition. Platform security enhances the security features of Symbian OS to deliver a more secure platform for mobile devices.

Two feature packs for S60 3rd Edition have been announced. These bring further enhancements to Symbian C++, Java™, and Flash Lite from Adobe developers, with new APIs that enable applications to offer exciting new features to users.

In S60 3rd Edition, Feature Pack 2, the presence of Web Run-Time allows S60 users to access Web 2.0 services and Internet content by using widgets. Using familiar Web-based technologies — such as HTML, CSS, JavaScript™, and Ajax — widgets allow developers to deliver Web 2.0 functionality to S60 devices more easily than ever.

The S60 platform provides developers with access to C++ APIs from Symbian OS and to the S60 UI and application engines.

S60 2nd Edition employs Symbian OS v7.0s and, in later feature packs, Symbian OS v8.0a and Symbian OS v8.1a. More than 30 sets of new APIs are made available in S60 2nd Edition, supporting features such as DRM, publish and subscribe, image manipulation, and scalable icons.

S60 3rd Edition provides industry-leading security underpinned by Symbian Signed and a new, more efficient binary format. C++ developers have access to new location, SIP, DRM, and IM APIs in this edition. S60 3rd Edition, Feature Pack 1 adds APIs for OCR, OpenGL V1.1, and improved interaction with platform features such as the gallery, settings, and phonebook. The latest addition, S60 3rd Edition, Feature Pack 2, adds support for Map framework APIs, RGA APIs, and APIs for seamless connection transition.

In addition, S60 3rd Edition includes Open C, which provides developers access to a set of standard POSIX and middleware C libraries. This access significantly improves developers'
ability to use existing C code and smooths the path for C developers who want to build applications for the S60 platform.

Java™ technology provides developers with a robust cross-platform alternative to C++. S60 2nd Edition added Java™ ME/MIDP 2.0 with enhanced performance from the CLDC HotSpot compiler. In addition, a number of new JSRs were implemented, including the Mobile 3D Graphics API (JSR-184), Java™ APIs for Bluetooth (JSR-82), and FileConnection API (JSR-75).


The Java implementation on the S60 platform shares many features with the implementations on the Series 40 platform and the Series 80 platform. This allows developers to easily extend their applications' reach to the entire range of platforms.

Flash Lite from Adobe became an optional feature of S60 3rd Edition in the form of support for Flash Lite 1.1. In Feature Pack 1, the supported version is Flash Lite 2.0, while Feature Pack 2 supports Flash Lite 2.1.

Flash Lite offers a development option for both content and applications. Flash Lite technology can deliver content or applications on S60 devices. The Web Browser for S60 can also play Flash Lite content offered on Web sites, and Flash Lite animations can be used as device screen savers from S60 3rd Edition, Feature Pack 1 onward.
Python developers now have access to a Python interpreter for all editions of the S60 platform. Whether prototyping applications for later C++ or Java™ implementation or delivering fully functional applications, Python provides a rapid-development environment for the S60 platform.

The S60 platform provides media developers with options to create browsing and messaging applications, deliver streaming audio and video, and create themes to personalize devices.

For browsing applications, S60 2nd Edition provides WML and XHTML Mobile Profile over WAP as well as support for HTTP/1.1 and browsing over TCP/IP. In addition, secure applications are possible through support for SSL and TLS. Browsing has been improved in S60 3rd Edition with the introduction of the Web Browser for S60. From S60 3rd Edition, Feature Pack 1 onward, the Web Browser for S60 is the platform’s default browser.

With S60 3rd Edition, Feature Pack 2, Web developers gain the ability to create widgets that deliver Web 2.0 functionality to S60 devices.

Developers of music-based applications and services for the S60 platform have access to an extensive range of APIs for manipulating and playing music. Those capabilities, coupled with the S60 platform’s state-of-the-art music player, afford developers extensive opportunities to create music-related applications and services. The S60 platform also provides support for streaming audio and video.

Messaging developers have the ability to use MMS through the SMIL support added to S60 2nd Edition. This allows messages to be created as animated slide shows.

Finally, S60 2nd Edition and S60 3rd Edition give media developers the ability to create themes for S60 devices, allowing customization of the look and sound of the user interface.
Screen resolution

The S60 platform supports several UI resolutions, as illustrated here:

<table>
<thead>
<tr>
<th>Portrait resolutions</th>
<th>Landscape resolutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>176 x 208 pixels</td>
<td>208 x 176 pixels</td>
</tr>
</tbody>
</table>

**Figure 3.5.** This "classic" S60 UI resolution has been supported from the first S60 device, the Nokia 7650 phone.

**Figure 3.6.** This resolution, introduced in S60 3rd Edition, offers a landscape version of the "classic" S60 UI resolution. This option has yet to be implemented in a device.

<table>
<thead>
<tr>
<th>240 x 320 pixels</th>
<th>320 x 240 pixels</th>
</tr>
</thead>
</table>

**Figure 3.7.** This UI, introduced in S60 2nd Edition, Feature Pack 3, offers a portrait QVGA format for S60 devices. Its first implementation was in the Nokia N92 device.

**Figure 3.8.** This UI, introduced in S60 3rd Edition, offers a landscape QVGA format for S60 devices. The first implementation was in the Nokia E61 smartphone.
Figure 3.9. This "double" UI, introduced in S60 2nd Edition, Feature Pack 3, offers four times the pixels of the "classic" S60 UI resolution. The first implementation was in the Nokia N90 device.

Figure 3.10. This UI, introduced in S60 3rd Edition, offers a landscape version of the "double" S60 UI resolution. The first implementation was in the Nokia N90 device, where it is used exclusively by the camera application. The Nokia E70 smartphone is the first device that offers this UI as a switchable (flip-open) option.
The following list shows the differences (new, modified, deprecated or removed) between S60 2nd and 3rd Edition in terms of APIs and architecture.

**NEW/MODIFIED**

- Real Time Kernel--EKA2 (3rd Edition, Symbian OS v9.1)
- Central Repository (3rd Edition, Symbian OS v9.1)
- Install system (3rd Edition, Symbian OS v9.1)
- New tool chain (New ARM RVCT and GCC EABI compilers) (3rd Edition, Symbian OS v9.1)
- Wide usage of ECOM plug-ins (3rd Edition, Symbian OS v9.1)
- Standard C++ additions
- On-target debugging (3rd Edition)
- Light API (3rd Edition)
- Vibra API (3rd Edition)
- Vibra Settings API (3rd Edition)
- HWRM Power State API (3rd Edition)
- Remote Control Framework (3rd Edition, Symbian OS v9.1)
- Profiles Engine Active Profile Settings API (3rd Edition)
- Find Item API (3rd Edition)
- Secure Digital (SD) card support (3rd Edition)
- Startup List Management API (3rd Edition)
- PtiEngine API (3rd Edition)
- WebUtils API (3rd Edition)
- Info Popup Note API (3rd Edition)
- Server Application API (3rd Edition)
- Common Audio Enhancements API (3rd Edition)
- Feature Discovery API (2nd FP3)
" Contacts Model API enhancement (2nd FP3, Symbian OS v8.1)
  • Publish & Subscribe API (2nd FP2, Symbian OS v8.0)
  • Message Queues (2nd FP2, Symbian OS v8.0)
  • XML Framework (2nd FP2, Symbian OS v8.0)
  • Huffman API (2nd FP2, Symbian OS v8.0)

REMOVED

  • System Agent (3rd Edition, Symbian OS v9.1)
  • Shared Data API (3rd Edition, Symbian OS v9.1)
  • CSettinglnfo API (3rd Edition)
  • Vibra Control API (3rd Edition)

The VibraCtrl API is replaced with a new, more robust Vibra API.

  • AIF (Application Information File) support (3rd Edition, Symbian OS v9.1)
  • PLPVariant API (3rd Edition, Symbian OS v9.1)

NEW

  • Platform security (3rd Edition, Symbian OS v9.1)
  • Content Access Framework (3rd Edition, Symbian OS v9.1)
  • OMA DRM CAF Agent API (3rd Edition)
  • DRM Audio Player API (3rd Edition)
  • OMA DRM v2.0 (3rd Edition, Symbian OS v9.1)
  • DRM Helper API (2nd FP2)
  • DRM3 License Checker API (2nd FP2)
  • IPSec/VPN Client Application (2nd FP2)

DEPRECATED

  • DRM Common API (2nd FP3)
This covers audio, video, graphics, imaging, and camera related.

NEW

- Exif API (3rd Edition)
- MIDI Client API (3rd Edition)
- DRM Audio Player API (3rd Edition)
- Audio Effects API (3rd Edition)
- Audio Effects Presets API (3rd Edition)
- Image Transform Library API (2nd FP3, Symbian OS v8.1)
- Camera Support Enhancements (2nd FP3)
- GIF Scaler API (2nd FP2, Symbian OS v8.0)
- Speech recognition (2nd FP2)
- 3-D Graphics — OpenGL ES (2nd FP2, Symbian OS v8.0)
- Enhanced Media Gallery (2nd FP2)
- eAAC+ and Other New Media Formats (3rd Edition)
- Streaming Rate Adaptation (3rd Edition)
- Streaming QoS (2nd FP3)

REMOVED

- Nokia Camera API—Camera Server (3rd Edition)

Communications

This covers networking (HTTP, sockets, TCP/IP, SIP), telephony, data bearers, local connectivity (IR, BT, serial cable, USB), and Web Services.

NEW

- Web Service Connection API (3rd Edition)
- Web Service Description API (3rd Edition)
- Web Service Manager API (3rd Edition)
- XML Extensions API (3rd Edition)
- SIP Stack and APIs (3rd Edition)
- SIP Client Resolver API (3rd Edition)
- OMA DNS 1.0 support (3rd Edition)
- WLAN support (3rd Edition)
- Network Status API (3rd Edition)
- 3rd Party Telephony API (3rd Edition, Symbian OS v8.0)
- Multiple secondary PDP contexts (2nd FP3)
- Connection Monitor API (2nd FP2)
- WCDMA support (2nd FP2)
- Circuit Switched Video Call (2nd FP2)
- EDGE support (2nd FP1)
- Bluetooth enhancements

Bluetooth 1.2 support, Remote SIM Access Profile (3rd Edition), Headset Profile, Basic Imaging Profile (2nd FP2), Bluetooth Notifier API (2nd FP1)

- OBEX (Object Exchange protocol services) API (2nd FP1)

REMOVED

- ETEL classes (3rd Edition, Symbian OS v9.1)
- RNiFManager / RGenericAgent (3rd Edition, Symbian OS v9.1)
- WAP Stack and WSP API (2nd FP3, Symbian OS v8.1)
- WAP Stack Client API (2nd FP2, Symbian OS v8.0)
- Bluetooth Registry API (2nd FP2, Symbian OS v8.0)
- PC Connectivity Server Framework API (2nd FP1)
- PLP Connection Link API (2nd FP1)
- Connectivity Framework API (2nd FP1)
NEW

* Qwerty keypad support (3rd Edition)
* Flash Lite 1.1 Viewer (3rd Edition)
* Scalable UI framework (2nd FP3)
* Scalable Icons API (2nd FP3)
* Pictograph API (2nd FP2)
* Semi-transparent windows (2nd FP2, Symbian OS v8.0)

MODIFIED

* S60 UI APIs (3rd Edition)

API optimizations are introduced; for example, all the virtual functions in SDK API classes that are derivable from outside the DLL are exported. S60 platform starts to use the Uikon UI library from Symbian, instead of its own version of it, which should improve UI compatibility and portability among Symbian OS-based platforms.

REMOVED

* Platform icons(avnkon.mbm) (3rd Edition)
* S60 UI localization file(avnkon.loc) (2nd FP2)

NEW

* Data Synchronization(ECOM) (3rd Edition)

REMOVED

* Certain SyncML APIs (3rd Edition)
NEW

- Landmarks API (3rd Edition)
- Landmark Search API (3rd Edition)
- Landmarks Database Management API (3rd Edition)
- Landmarks UI Add/Edit API (3rd Edition)
- Landmarks UI Selector API (3rd Edition)
- BLID Application Satellite Info API (3rd Edition)
- Location Acquisition API (2nd FP2)

NEW

- Download Manager APIs (2nd FP3/3rd Edition)
- OMA Download Support for Billing Notification (3rd Edition)
- Browser Control API (2nd FP3)
- Browser Plug-in API (2nd FP2)
- Large File Downloads over HTTP (2nd FP3)
- Plug-in Browser for (X)HTML Rendering
- XHTML Mobile Profile 1.1 (2nd FP2)
- HTTP Digest Authentication (2nd FP2)
- Content/File Upload (2nd FP2)
- Links to Other Applications (2nd FP2)
- New Accept Header (2nd FP2)
- SMS, MMS, and MMSto Schemes (2nd FP2)
- ECMAScript Mobile Profile (2nd FP1)
- Support for HTML 4.01 (2nd FP1)
- Meta and Object Tag (2nd FP1)
- Performance Enhancements (2nd FP1)
- UI Enhancements
  - Enhanced narrow-screen rendering (2nd FP3)
Scalable UI support (2nd FP3)
Frames (2nd FP2)
Full-screen mode (2nd FP1)
A download progress bar (2nd FP1)
Scrollbars (2nd FP1)
Adaptive history list (2nd FP1)
An auto-complete feature (2nd FP1)
Background Image (2nd FP1)

Push Service Enhancements (2nd FP1)

REMOVED

Wallet Application Is Removed (2nd FP3)

NEW

New Message Notification API (3rd Edition)

SMS Enhancements

Support for WCDMA Cell Broadcast (CBS) (3rd Edition)
Showing the unused characters real-time (3rd Edition)
SMS and Internet electronic mail interworking (3rd Edition)
Reading and selecting SMSC (3rd Edition)
Sending and receiving of SMS messages in WCDMA (2nd FP2)
Highlighted items in the message viewer (2nd FP2)
MMS Enhancements

- Subject, recipient, and priority fields (3rd Edition)
- Enhanced delivery reports (3rd Edition)
- MMS Postcard client (3rd Edition)
- MMS Upload services (2nd FP3)
- MMS Enhancements in 2nd Edition FP2

Email Enhancements

- Subject and recipient fields (3rd Edition)
- E-mail enhancements in 2nd Edition FP2
- Enhancements in 2nd Edition FP1 (use of port number 995 for encrypted connections over SL in POP for Microsoft Exchange Server.)

Instant Messaging and Presence

- IM enhancements in 3rd Edition
- IM API (3rd Edition)
- IM Application Launch API (3rd Edition)
- IM and Presence enhancements in 2nd Edition FP2
- Instant Messaging and Presence (2nd FP1)
- Chat application (2nd FP1)

REMOVED

- Presence API (3rd Edition)

NEW

- Local time and daylight saving support (2nd FP3, Symbian OS v8.1)
Time zone is supported by the S60 platform’s own clock and calendar applications from 3rd Edition onwards, but the APIs provided by Symbian OS v8.1 should be available for 3rd-party developers already in S60 2nd Edition, Feature Pack 3.

- Location API, JSR-179 (3rd Edition)
- SIP API, JSR-180 (3rd Edition)
- Security and Trust Services, JSR-177 (3rd Edition)
- WMA API 2.0, JSR-205 (3rd Edition)
- Scalable 2D Vector Graphics API, JSR-226 (3rd FP1)
- Effects of Scalable UI for Java MIDP (2nd FP3)
- J2ME Web Services Specification, JSR-172 (2nd FP3)
- OBEX support (2nd FP3)
- Push registry support (2nd FP2)
- CLDC 1.1, JSR-139 (2nd FP2)
- PIM API, JSR-75 (2nd FP2)
- FileConnection API, JSR-75 (2nd FP2)
- Mobile 3D Graphics API, JSR-184 (2nd FP2)
- Complete JTWI 1.0 compliance, JSR-185 (2nd FP2)
- Enhancements for Mobile Media API, JSR-135
- Nokia UI API is deprecated (2nd Edition)
- Platform security

NEW

- OMA DRM v2.0 (3rd Edition)
- OMA DRM v1.0 (2nd FP2)

From S60 2nd Edition, Feature Pack 2 onwards the S60 platform supports the full OMA DRM v1.0 (forward-lock, combined delivery, and separate delivery methods).
Symbian OS software written for the S60 Platform 1st edition (S60v1) or 2nd edition (S60v2) is not binary compatible with S60 3rd edition (S60v3), because it uses a new, hardened version of the Symbian OS (v9.1).

Series 60 1st Edition (Version 1.0), based on Symbian OS 6.1

- Nokia 7650 (development name Calypso)

Series 60 1st Edition (Version 1.2), based on Symbian OS 6.1

- Nokia 3600
- Nokia 3620 (GSM 850/1900 successor of the 3650) (Shrek)
- Nokia 3650 (Cameron)
- Nokia 3660 (GSM 900/1800/1900 successor of the 3650) (Sylvester)
- Nokia N-Gage (The Starship)
- Nokia N-GageQD
- Samsung SGH-D700
- Sendo X
- Sendo X2
- Siemens SX1

Series 60 2nd Edition (Version 2.0), based on Symbian OS 7.0s

Contains new features like J2ME/MIDP 2.0 support and themes

- Nokia 6600 (development name Calimero)
- Samsung SGH-D710
- Samsung SGH-D720
- Samsung SGH-D730

Series 60 2nd Edition, Feature Pack 1 (Version 2.1), based on Symbian OS 7.0s

- Nokia 3230 (Blitz)
- Nokia 6260
- Nokia 6620
- Nokia 6670
- Nokia 7610 (development name Catalina)
- Panasoninc X700
Panasonic X800

Series 60 2nd Edition, Feature Pack 2 (Version 2.6), based on Symbian OS 8.0a

Symbian 8.x has support for WCDMA (GSM 3G) and cdma2000 networks.

- Lenovo P930
- Nokia 6630 (development name Charlie)
- Nokia 6680 (Milla)
- Nokia 6681 (Cho)
- Nokia 6682

Series 60 2nd Edition, Feature Pack 3 (Version 2.8), based on Symbian OS 8.1a

- Nokia N70
- Nokia N72
- Nokia N90 (Gromit)
- Samsung SGH-Z600

S60 3rd Edition (Version 3.0), based on Symbian OS 9.1

- Nokia 3250 (Thunder)
- Nokia 5500
- Nokia E50
- Nokia E60
- Nokia E61
- Nokia E61i
- Nokia E62
- Nokia E65
- Nokia E70
- Nokia N71
- Nokia N73
- Nokia N75
- Nokia N77
- Nokia N80
- Nokia N91 (Nemo)
- Nokia N92
- Nokia N93
Nokia N93i

S60 3rd Edition, Feature Pack 1 (Version 3.1), based on Symbian OS 9.2

- LG KS10 JoY
- Nokia 5700 XpressMusic
- Nokia 6110 Navigator
- Nokia 6120 Classic
- Nokia 6121 Classic
- Nokia 6290
- Nokia E51
- Nokia E90 Communicator
- Nokia N76
- Nokia N81
- Nokia N81 8GB
- Nokia N82
- Nokia N95
- Nokia N95 8GB
- Nokia N95-3 NAM
- Samsung SGH-i400
- Samsung SGH-i520
- Samsung SGH-i550

**S60 Platform and device identification codes**

Below shows S60 identification codes. There are three types of identification codes, i.e.:

- Platform ID
- Product ID
- Machine UID

The Platform ID and Product ID are usually used in the package file (.pkg) to perform checking during installation. For example, an installation file can display warning if a user is trying to install it on incompatible platform or device.
The Machine UID is usually used in the code during runtime. The API to get Machine UID in C++ is `HAL::Get(HAL::EMachineUid)`. It is also possible to check Machine UID during installation using IF statement in the .pkg file.

<table>
<thead>
<tr>
<th>PlatformID</th>
<th>S60 version</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0x101F6F88)</td>
<td>S60 1st Edition (0.9)</td>
</tr>
<tr>
<td>(0x101F8202)</td>
<td>S60 1st Edition, FP1</td>
</tr>
<tr>
<td>(0x101F7960)</td>
<td>S60 2nd Edition</td>
</tr>
<tr>
<td>(0x101F9115)</td>
<td>S60 2nd Edition, FP1</td>
</tr>
<tr>
<td>(0x10200BAB)</td>
<td>S60 2nd Edition, FP2</td>
</tr>
<tr>
<td>(0x102032BD)</td>
<td>S60 2nd Edition, FP3</td>
</tr>
<tr>
<td>[0x101F7961]</td>
<td>S60 3rd Edition</td>
</tr>
<tr>
<td>[0x102032BE]</td>
<td>S60 3rd Edition, FP1</td>
</tr>
<tr>
<td>[0x102752AE]</td>
<td>S60 3rd Edition, FP2</td>
</tr>
</tbody>
</table>

**S60 1st Edition**

<table>
<thead>
<tr>
<th>Product ID</th>
<th>Machine UID</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x101F6F87</td>
<td>0x101F4FC3</td>
<td>Nokia 7650</td>
</tr>
</tbody>
</table>

**S60 1st Edition, FP1**

<table>
<thead>
<tr>
<th>Product ID</th>
<th>Machine UID</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x101F7962</td>
<td>0x101F466A</td>
<td>Nokia 3650</td>
</tr>
</tbody>
</table>
### Analysis of Wireless Devices Platform and Professional Developing in J2ME Applications

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>0x101F466A</td>
<td>Nokia 3660</td>
</tr>
<tr>
<td>N/A</td>
<td>0x101FA031</td>
<td>Sendo-X</td>
</tr>
<tr>
<td>0x101F9071</td>
<td>0x101F9071</td>
<td>Siemens SX-1</td>
</tr>
</tbody>
</table>

### S60 2nd Edition

<table>
<thead>
<tr>
<th>Product ID</th>
<th>Machine UID</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x101F7963</td>
<td>0x101FB3DD</td>
<td>Nokia 6600</td>
</tr>
</tbody>
</table>

### S60 2nd Edition, FP1

<table>
<thead>
<tr>
<th>Product ID</th>
<th>Machine UID</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x10200F97</td>
<td>0x10200F97</td>
<td>Nokia 3230</td>
</tr>
<tr>
<td>0x101FB3F4</td>
<td>0x101FB3F4</td>
<td>Nokia 6260</td>
</tr>
<tr>
<td>0x1020216B</td>
<td>0x101F3EE3</td>
<td>Nokia 6620</td>
</tr>
<tr>
<td>0x101FD5DC</td>
<td>0x101FB3F3</td>
<td>Nokia 6670</td>
</tr>
<tr>
<td>0x101FD5DB</td>
<td>0x101FB3F3</td>
<td>Nokia 7610</td>
</tr>
</tbody>
</table>

### S60 2nd Edition, FP2

<table>
<thead>
<tr>
<th>Product ID</th>
<th>Machine UID</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x101F7964</td>
<td>0x101FBB55</td>
<td>Nokia 6630</td>
</tr>
<tr>
<td>0x102078D1</td>
<td>0x10200F99</td>
<td>Nokia 6680</td>
</tr>
<tr>
<td>0x102078D0</td>
<td>0x10200F9C</td>
<td>Nokia 6681</td>
</tr>
<tr>
<td>0x102078CF</td>
<td>0x10200F9B</td>
<td>Nokia 6682</td>
</tr>
</tbody>
</table>
### S60 2nd Edition, FP3

<table>
<thead>
<tr>
<th>Product ID</th>
<th>Machine UID</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x10200F9A</td>
<td>0x10200F9A</td>
<td>Nokia N70</td>
</tr>
<tr>
<td>0x10200F9A</td>
<td>0x10200F9A</td>
<td>Nokia N72</td>
</tr>
<tr>
<td>0x10200F98</td>
<td>0x10200F98</td>
<td>Nokia N90</td>
</tr>
</tbody>
</table>

### S60 3rd Edition

<table>
<thead>
<tr>
<th>Product ID</th>
<th>Machine UID</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x200005F8</td>
<td>0x200005F8</td>
<td>Nokia 3250</td>
</tr>
<tr>
<td>0x20000602</td>
<td>0x20000602</td>
<td>Nokia 5500 Sport</td>
</tr>
<tr>
<td>0x20002495</td>
<td>0x20002495</td>
<td>Nokia E50</td>
</tr>
<tr>
<td>0x20001856</td>
<td>0x20001856</td>
<td>Nokia E60</td>
</tr>
<tr>
<td>0x20001858</td>
<td>0x20001858</td>
<td>Nokia E61</td>
</tr>
<tr>
<td>0x20002D7F</td>
<td>0x20002D7F</td>
<td>Nokia E61i</td>
</tr>
<tr>
<td>0x20001859</td>
<td>0x20001859</td>
<td>Nokia E62</td>
</tr>
<tr>
<td>0x20000604</td>
<td>0x20000604</td>
<td>Nokia E65</td>
</tr>
<tr>
<td>0x20001857</td>
<td>0x20001857</td>
<td>Nokia E70</td>
</tr>
<tr>
<td>0x200005FF</td>
<td>0x200005FF</td>
<td>Nokia N71</td>
</tr>
<tr>
<td>0x200005FB</td>
<td>0x200005FB</td>
<td>Nokia N73</td>
</tr>
<tr>
<td>0x200005FE</td>
<td>0x200005FE</td>
<td>Nokia N75</td>
</tr>
<tr>
<td>0x20000601</td>
<td>0x20000601</td>
<td>Nokia N77</td>
</tr>
<tr>
<td>0x200005F9</td>
<td>0x200005F9</td>
<td>Nokia N80</td>
</tr>
<tr>
<td>0x200005FC</td>
<td>0x200005FC</td>
<td>Nokia N91</td>
</tr>
<tr>
<td>0x200005FA</td>
<td>0x200005FA</td>
<td>Nokia N92</td>
</tr>
</tbody>
</table>
## S60 3rd Edition, FP1

<table>
<thead>
<tr>
<th>Product ID</th>
<th>Machine UID</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x20002D7C</td>
<td>0x20002D7C</td>
<td>Nokia 5700 XpressMusic</td>
</tr>
<tr>
<td>0x20002D7B</td>
<td>0x20002D7B</td>
<td>Nokia 6110 Navigator</td>
</tr>
<tr>
<td>0x20002D7E</td>
<td>0x20002D7E</td>
<td>Nokia 6120 Classic</td>
</tr>
<tr>
<td>0x20000606</td>
<td>0x20000606</td>
<td>Nokia 6290</td>
</tr>
<tr>
<td>0x20002498</td>
<td>0x20002498</td>
<td>Nokia E51</td>
</tr>
<tr>
<td>0x20002496</td>
<td>0x20002496</td>
<td>Nokia E90</td>
</tr>
<tr>
<td>0x2000060A</td>
<td>0x2000060A</td>
<td>Nokia N76</td>
</tr>
<tr>
<td>0x20002D83</td>
<td>0x20002D83</td>
<td>Nokia N81 &amp; N81 8GB</td>
</tr>
<tr>
<td>0x20002D85</td>
<td>0x20002D85</td>
<td>Nokia N82</td>
</tr>
<tr>
<td>0x2000060B</td>
<td>0x2000060B</td>
<td>Nokia N95 &amp; N95-3 NAM</td>
</tr>
<tr>
<td>0x20002D84</td>
<td>0x20002D84</td>
<td>Nokia N95 8GB</td>
</tr>
</tbody>
</table>

## N-Gage

<table>
<thead>
<tr>
<th>Product ID</th>
<th>Machine UID</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x101F8A64</td>
<td>0x101F8C19</td>
<td>Nokia N-Gage™</td>
</tr>
<tr>
<td>N/A</td>
<td>0x101FB2B1</td>
<td>Nokia N-Gage™ QD</td>
</tr>
</tbody>
</table>