



Contents

- [1 Introduction](#)
- [2 S60 platform overview](#)
 - ◆ [2.1 S60 platform architecture](#)
- [3 Key changes on S60 3rd Edition](#)
 - ◆ [3.1 ABI compiler for the ARM architecture](#)
 - ◆ [3.2 Real-time kernel](#)
 - ◆ [3.3 Platform security](#)
- [4 S60 products](#)
- [5 S60 application development](#)
 - ◆ [5.1 Installations](#)
 - ◇ [5.1.1 Register a Forum Nokia account](#)
 - ◇ [5.1.2 Install development tools](#)
 - ◇ [5.1.3 Install S60 3rd Edition SDKs](#)
 - ◇ [5.1.4 Install Java runtime environment](#)
 - ◇ [5.1.5 Install Perl](#)
 - ◆ [5.2 Development process for S60 3rd Edition](#)
 - ◆ [5.3 Deployment of S60 3rd Edition](#)
- [6 S60 support and useful links](#)
 - ◆ [6.1 S60 support](#)
 - ◆ [6.2 Useful links](#)
- [7 Conclusions](#)

Introduction

The S60 platform is the world's leading smartphone software platform, offering a feature-rich software base for phones with advanced data capabilities. The S60 platform supports multitasking between applications and provides the user interfaces for switching between running applications and starting new ones while others are running.

This document is targeted at application developers, content providers, system architects, specification writers, requirement managers, product managers, and those who either use the S60 platform directly or plan for applications using its features.

This document is an introduction to the S60 platform. It mainly focuses on the some of the concepts elaboration regarding S60 platform. Meanwhile, it gives a whole introduction on how to develop applications on Nokia S60 platform. Especially, the emphasis is given to the development of S60 3rd Edition, which is the latest S60 platform, and it differs a lot from its predecessors in terms of development process and platform security.

S60 platform overview

The S60 platform is the market-leading smartphone platform built on Symbian OS. It incorporates all key mobile technologies expected by increasingly sophisticated enterprise users and consumers, and it provides revenue opportunities for the full range of stakeholders in the mobile marketplace. As the S60 platform has developed, it has raised the bar on smartphone feature provision by taking the lead in developing and implementing many innovations.

S60 platform architecture

The following figure illustrates the high level of architecture of the S60 platform. The platform is based on Symbian OS but also provides additional features:

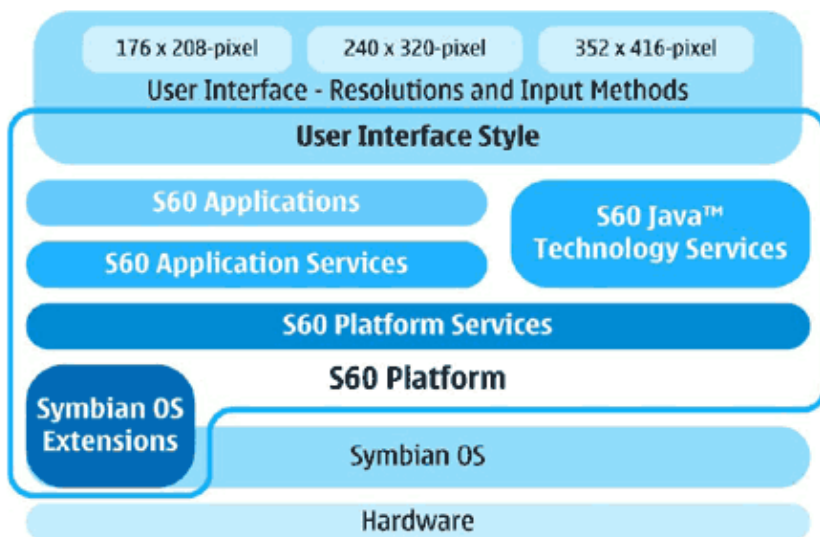


Figure 1: S60 platform architecture

As shown in the above figure, S60 platform is based on the Symbian OS, which evolves all the time. As such, S60 platform has different editions. S60 platform has experienced 1st Edition, 2nd Edition, and the latest 3rd Edition. In each of the edition, it also introduces different feature packs, which incorporate some of the advanced features on each release. Symbian OS is based on open standard, which makes the development on Symbian/S60 open for the developers. The Development for the S60 platform is backward compatible although there was a break between the 2nd Edition and the 3rd Edition, which was introduced by the platform security and new compiler used in the platform.

The releases and their respective operating systems are:

- **S60 1st Edition** ? Symbian OS v6.1.
- **S60 2nd Edition** ? Symbian OS v7.0s.
 - ◆ S60 2nd Edition, Feature Pack 1 ? Symbian OS v7.0s.

S60_3rd_Edition:_Application_Development

- ◆ S60 2nd Edition, Feature Pack 2 ? Symbian OS v8.0a.
- ◆ S60 2nd Edition, Feature Pack 3 ? Symbian OS v8.1a.
- **S60 3rd Edition** ? Symbian OS v9.1.
 - ◆ S60 3rd Edition, Feature Pack 1 ? Symbian OS v9.2.
 - ◆ S60 3rd Edition, Feature Pack 2 ? Symbian OS v9.3.
- **S60 5th Edition** ? Symbian OS v9.4.

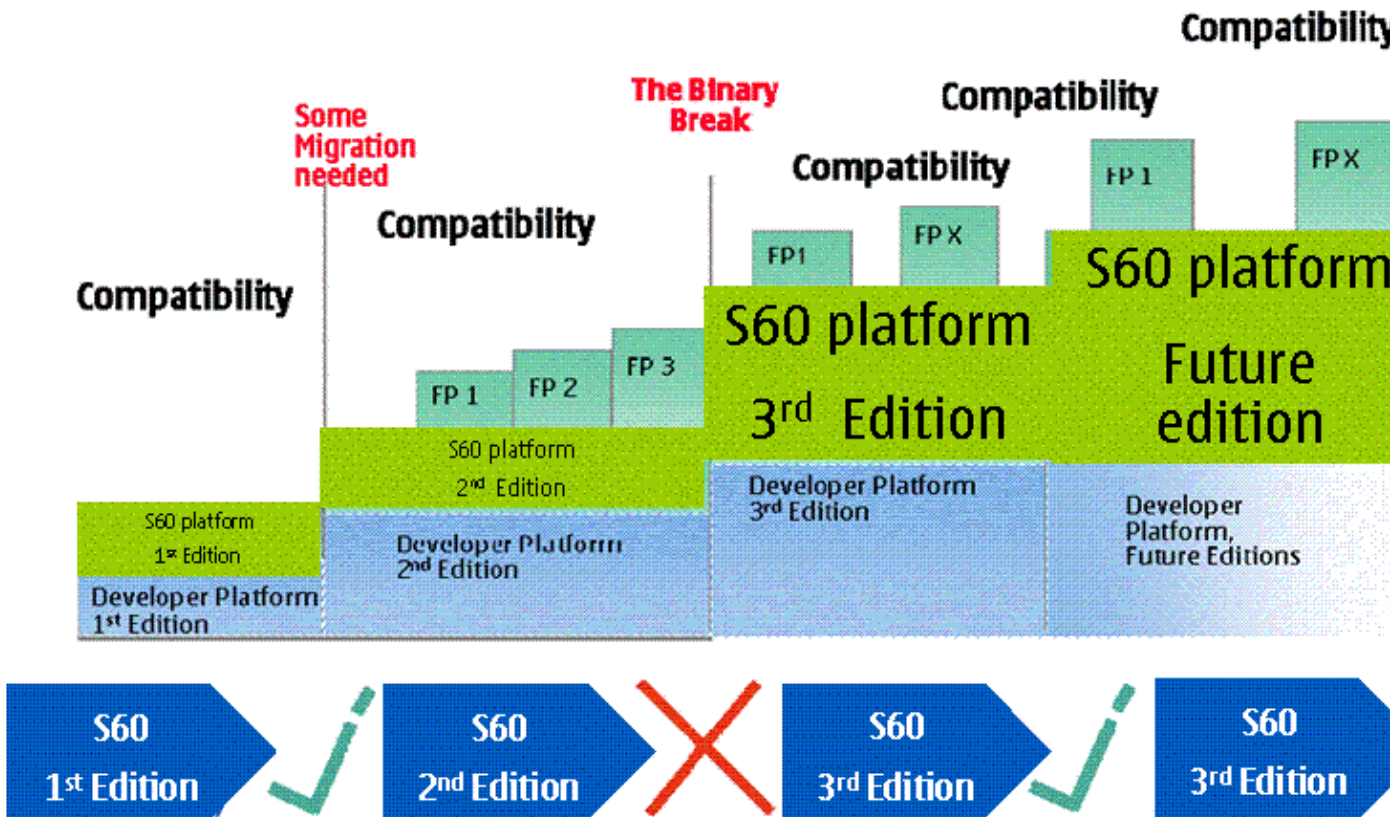


Figure 2?S60 platform evolution

Before starting development, developers may refer to the Forum Nokia Web site (www.forum.nokia.com/documents) to check feature details of the Symbian OS version that is incorporated in each edition or feature pack. A developer may download the document **S60 3rd Edition: What's new for developers** to find out the changes on S60 3rd Edition SDK and the main APIs introduced in each of the feature packs. More information regarding Symbian OS can be found at www.symbian.com.

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.

S60 platform services are the fundamental services provided by the S60 platform, these include:

- 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.

S60_3rd_Edition:_Application_Development

- 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.

S60 application services a set of capabilities that are used by the S60 applications and can be used by third-party developers to provide basic functionality for applications. These include ?PIM Application Services,? ?Messaging Application Services,? and ?Browser Application Services.?

S60 Java technology services support 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 configuration (JSR-139) and the Mobile Information Device Profile (MIDP) 2.0 (JSR-118) extension to this configuration. In addition, some of the additional APIs are supported as well.

S60 Applications, applications available to a device?s user, include personal information manager (PIM), messaging, media applications, profiles, etc.

The S60 platform defines a **UI style and APIs**, but it does not mandate the screen size or input methods. Licensees are free to implement their own customized UIs. Developers must program UI applications with scalability in mind because specific UI dimensions cannot be assumed. Currently, since the introduction of Scalable UI, S60 supports 3 different screen sizes: 176x208 pixels, 240x320 pixels, 352x416 pixels, and the display can be in landscape or portrait orientation. A developer only needs to write once (taking care of the Scalable UI), then application can be applicable to different screen sizes.

For further information regarding S60, visit the following Web sites:

<http://www.s60.com>

<http://www.forum.nokia.com>

<http://www.forum.nokia.com.cn> (Chinese)

Key changes on S60 3rd Edition

On S60 3rd Edition, there are some big changes, and some important concepts are introduced. It is very important to understand these concepts before doing some development on S60 3rd Edition.

In short, the biggest differences between S60 2nd Edition and S60 3rd Edition are:

- **ABI compiler for the ARM architecture**
- **Real Time Kernel (from Symbian OS v9 onwards)**
- **Platform security**
- **Symbian Signed**

- **Tools**

ABI compiler for the ARM architecture

From S60 3rd Edition onwards, a compiler based on the Application Binary Interface (ABI) for the ARM® Architecture is used to compile applications. The ARM C/C++ ABI is an industry standard that determines how executables and shared objects work together. 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. Currently, there are two compilers: **GCCE** and **ARMV5**. The **GCCE** compiler comes with the SDK installation, and it is totally free. The **ARMV5** compiler is a commercial compiler, and a developer may find more information at ARM for pricing!

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).

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 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.

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.

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 (unless the application has enough capabilities). 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. In order to access the specific directories, an application needs to have some capabilities. The capability requirements for each of the specific directories in the device are shown as follows:

Directories (inc. sub directories)	Required Capabilities for:	
	Read Access	Write Access
/resource	None	TCB
/sys	AllFiles	TCB
/private/<ownSID>	None	None
/private/<otherSID>	AllFiles	AllFiles
/<other>	None	None

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 consumer (that is, the person who installs the application on the device) 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 to All** ? These capabilities are open to all applications; applications do not need to be Symbian Signed.
- ◇ **Granted by the User at Installation Time** ? 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.
- ◇ **Granted Through Symbian Signed** ? These capabilities include the *Granted by the User at Installation Time* capabilities and an extended set that require that an application be Symbian Signed before the capabilities can be accessed.

- ◇ **Granted Through the Manufacturer** ? These capabilities require an agreement with the device manufacturer to allow the capabilities to be granted during the Symbian Signed process.

Different capabilities need to have different types of certificates to develop. During the development phase, a user may need to apply for different certificates for testing. A certificate can be used to sign any of the applications, which has the subset of capabilities defined in the certificate. The S60 capabilities are described in the following picture:

Symbian signed and **Tools** will be introduced in the later parts of this article.

For more details on the device features associated with each capability, see the Forum Nokia document [Symbian OS: Overview to Security](#)

For a list of known sensitive applications that do need device manufacturer approval, see the page on [sensitive applications](#).

S60 products

The S60 platform is licensed by a number of stakeholders that have developed devices for the marketplace ? Lenovo, LG Electronics, Nokia Mobile Phones, Panasonic and Samsung. The range of S60 devices is constantly changing, and detailed information on the current range of devices can be found in the Devices section of the S60 Web site and the following sites:

[S60 devices](#)

[Nokia S60 devices](#)

[Symbian devices](#)

S60 application development

For new developers, developing on S60 3rd Edition could be difficult to some extent, and sometimes it is hard to know where to start. In this section, the development steps for the S60 3rd Edition are introduced. The steps are a little different from the previous platforms (1st and 2nd Editions) in terms of deployment onto a real device. The development on 3rd Edition is a little bit trickier than on the previous platforms. In the following, all of the relevant steps are described in detail.

Installations

Register a Forum Nokia account

For many of the resources at the Forum Nokia Web site, you need to register before downloading. The registration is free, and all of the SDKs, documents, and sample codes are free to download. The Forum Nokia registration Web site is:

<http://www.forum.nokia.com/main/registration/registration.html>

Install development tools

Currently, there are a number of development tools available for developing S60 3rd Edition. They are:

Please note that, following the withdrawal of Carbide.vs, Forum Nokia no longer supports the Microsoft Visual Studio IDE for Symbian C++ development. For more information see the [Carbide.vs withdrawal page](#) on Forum Nokia.

◇ Microsoft Visual Studio .NET 2003/2005

You need to install the Microsoft Visual Studio .NET 2003/2005 first. Then go to the [tools](#) section of the Forum Nokia Web site to download [Carbide.vs 2.0.2/Carbide.vs 3.01](#) (the download is free). A developer may also purchase the tool [Visual Assist X](#) to achieve better user experience.

If a developer still cannot find the **S60 3rd Edition FP2** SDK after having successfully installed the **S60 3rd Edition FP2** SDK and [Carbide.vs 2.0.2/Carbide.vs.3.01](#), please refer to the wiki article: [S60 3rd Edition SDK FP2 - Workarounds and Updates](#)

◇ Metrowerks Codewarrior

This will be a discontinued product from Nokia (Nokia owns the product), and it will be gradually phased out. It will be replaced by a new development tool called **Carbide. c++**. You can download the Professional, Personal, or OEM version from the [tools](#) section of the Forum Nokia Web site. After the installation, you have a trial period before it expires. The tool supports on-device debugging. In order to support on-device debugging, you have to install the following CodeWarrior update:

[CodeWarrior On-Device Debug Kit for Series 60 3rd Edition](#)

You can download the video file [On-Device Debugging for Symbian OS C++ Applications](#) from the Forum Nokia Web site.

The user guide [On-Device Debugging for Symbian OS Software](#) is also available at Forum Nokia.

Note: If you want to use the commands:

```
bldmake bldfiles
abld build winscw udeb
abld build gcce urel
```

to build your application, download CodeWarrior and install it. Otherwise, the commands may not work properly. Without a license, the Codewarrior IDE may not be started; however, the commands should be working anyway! When building an application in MS-DOS box, make sure the **EPOCROOT** environment variable is set properly. More information regarding the EPOCROOT setting can be found in SDK help. Alternatively, without using **EPOCROOT**, you are recommended to use the **devices** command to select the needed SDK if there are several SDKs installed in the PC. You can type the following command in MS-DOS box to get more information about the **devices** command:

devices --help

◇ **Carbide C++ Express 1.0/1.1/1.2/1.3/2.0**

This is the only free tool available for S60 3rd Edition development. You can freely download the tool from the Forum Nokia Web site at <http://www.forum.nokia.com/carbide>.

◇ **Carbide.c++ 1.0/1.1/1.2/1.3/2.0**

This is the latest development tool available from Nokia, and it supports the UIQ and S60 platforms. You can find more information about the tool from the Forum Nokia Web site, and you can have a free trial period before committing a real purchase.

The tool supports on-device debugging and has a wonderful UI designer inside the tools, which speeds up the development of Symbian applications. Meanwhile, for "Professional" and "OEM" version, it provides more advanced features for benchmarking and debugging.

A video [Carbide On-Device Debug Screencast](#) is available at Forum Nokia. You can download [Carbide.c++ Developer and Professional Edition, v1.1](#) from Forum Nokia.

If a developer wants to purchase the development tool Carbide.c++, please visit the website [Carbide sales contacts](#).

For Carbide.c++ 1.3 tool, a developer may also download the [C++ Developer's Library](#).

A developer may follow the following steps to configure to use **MS-DOS** box to compile a project using command line:

Start ==> Programs ==> Carbide.c++ 1.3 ==> Configure environment for WinsCW command line

It is recommended to install the tools before installing the S60 3rd Edition SDK since SDK installation may update some of the components in the corresponding tools.

Install S60 3rd Edition SDKs

You can download the S60 3rd Edition SDKs from <http://www.forum.nokia.com/tools> (section "C++ for Symbian OS Tools and SDKs"). A number of S60 SDKs are available. You can download the following SDKs for S60 3rd Edition application development:

- **3rd Edition**
- **3rd Edition, Maintenance Release**
- **3rd Edition, FP1**
- **3rd Edition, FP2**

There is also a "3rd Edition" release, which was the first release of S60 3rd Edition. If you want to develop an application that runs on all S60 3rd Edition devices including those from FP1/FP2 onwards, it is

S60_3rd_Edition:_Application_Development

recommended to download the maintenance SDK for development since this is a mature release and the base for all the future S60 3rd Edition SDKs (although future feature packs may introduce some advanced features).

If a developer still cannot find the **S60 3rd Edition FP2** SDK after having successfully installed the **S60 3rd Edition FP2** SDK and "Carbide.vs 2.0.2/Carbide.vs.3.01", please refer to the wiki article: [S60 3rd Edition SDK FP2 - Workarounds and Updates](#)

If a developer installs **3rd Edition, FP1** SDK, it is recommended to install [NCNList_fix.zip](#) to avoid the Application closed: ncncnlist KERN-EXEC 3" panic message when emulator is launched!

For Chinese developers, there is no separate Chinese SDK although it was the case in the previous edition SDK releases. Chinese language has been integrated into the European SDK. You only need to enable it by completing the following step in your computer:

Start ==> Programs ==> S60 Developer Tools ==> 3rd Edition SDK ==> 1.1 MR ==> Languages ==> Change to Chinese

However, it seems not so straightforward to make the Chinese/other languages input method working by selecting the # key on the emulator as normally is done on the real phone.

A workaround to solve to problem is to select the **ABC** key (on which has a pen icon). When an editor control is shown (for example, when composing an SMS message), selecting the **ABC** key will show up a menu to select the wanted input method. Once the input method is selected, a developer may use mouse to select the buttons (**1-9**) to type in Chinese or other selected language characters.

Please be noted, once S60 3rd Edition SDK is installed, all of the future project files should be resided in the same drive as the one where S60 SDK is installed. Meanwhile, the installation path should not contain any space characters. Normally, the default installation path is accepted!

Plug-ins:

- You can install the following plug-in for the Maintenance SDK for better multimedia emulation: [Extensions plug-in for S60 3rd Edition SDK for Symbian OS, for C++, MR or SDK API Plug-in](#)
- You can also install the [S60 3rd Edition: Device Management Plug-in](#)
- A [Sensor plug-in](#) is available for the Nokia 5500 sport phone.
- Design S60 application using Open C/C++ language [S60 Open C plug-in](#)". Now, with the latest release of the Open C/C++ package, a developer may use C/C++ languages to develop their games using the **RGAA** (Realtime Graphics and Audio APIs) software components in the package.

If the SDK installation fails, take a look at the following post on the Forum Nokia Web site: [Can't install S60 3rd SDK](#). It provides a workaround solution for solving the problem. You may need to create an empty file

S60_3rd_Edition:_Application_Development

"S60_3rd.xml", and put it into the directory where the 3rd Edition SDK installation package is unzipped.

If you install and uninstall the S60 3rd Edition maintenance SDK, the folder name of the installation will change (for example, \Symbian\9.1\S60_3rd_MR_1 or \Symbian\9.1\S60_3rd_MR_2). To remove the installation sequence number **1** or **2**, you can remove the following registry from the Windows registry database in your PC before installing:

HKEY_LOCAL_MACHINE\SOFTWARE\Nokia\com.nokia.s60

If you want to build your own application using command, refer to the SDK help for more information. Sometimes, on one computer, a few SDKs are installed. To ease the effort switching from one SDK to another, use the tool **environmentswitch.exe**. The tool is unfortunately not available on the 3rd Edition SDK. You can find this tool in the previous SDK installation. For example,

\Symbian\8.1a\S60_2nd_FP3\Series60Tools\environmentswitch\environmentswitch.exe

If a developer does not have the SDK installed, please go to the link [S60 3rd Edition Application Development](#) to download it (**EnvironmentSwitch.zip**).

Install Java runtime environment

Download the latest Java SDK and runtime environment from the following Web site: <http://java.sun.com>. Make sure all the needed paths and environment variables are correctly added to the computer environment variables using "Control Panel".

Install Perl

Find the latest version of Perl at the following Web site:

[ActivePerl-5.6.1.635](#)

Make sure to set the correct path of the executables in the **Control Panel** of your PC. Please do install the correct perl version according to the S60 3rd Edition SDK release notes. Do not need to install the latest version of the perl. Otherwise, it may have compatibility problems. Some developers do meet the problems after installing the latest perl.

Development process for S60 3rd Edition

The development process on S60 3rd Edition is a little different than on S60 2nd Edition due to the introduction of the platform security model on S60 3rd Edition and the new compiler used. The following figure describes the development process on S60 3rd Edition in a nutshell.

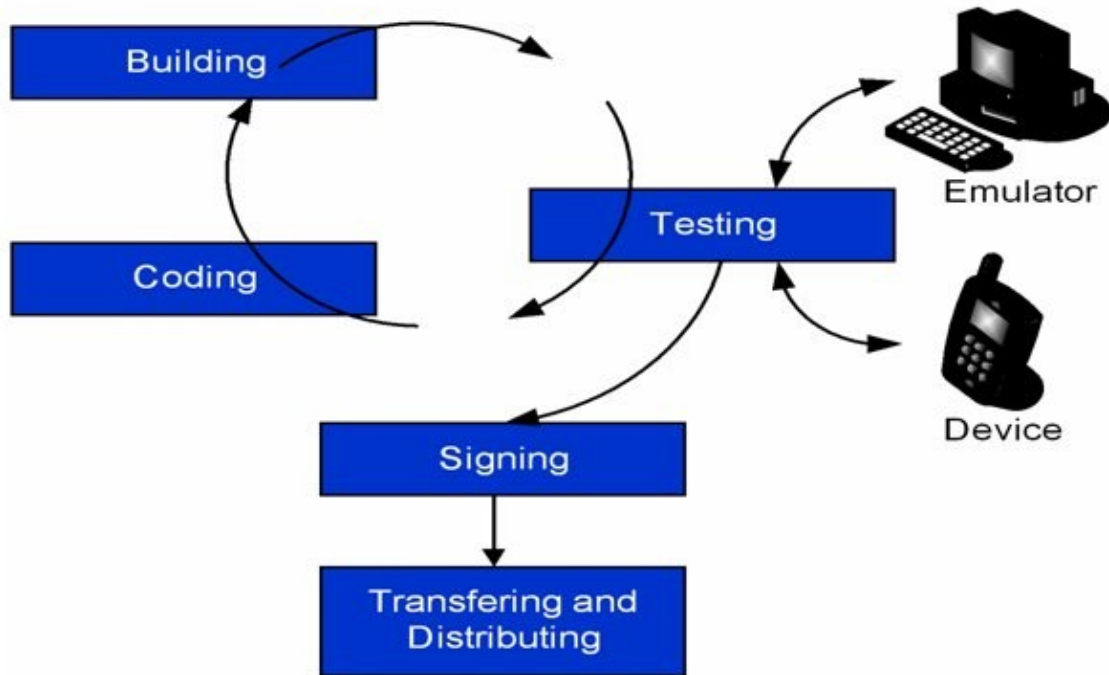


Figure 3: S60 application development process

The whole process is described in the document [S60 3rd Edition: Tool Chain, IDEs, And Development Process](#)

As shown in figure above, the steps can be described as follows:

1. Coding

You can start with an existing project by importing an .mmp file, or use the "Wizard" in the IDE tool to generate a template application. Normally, this is the recommended approach.

For those greenhorn Symbian OS C++ developers, Symbian OS C++ programming is a little bit hard to understand compared to ANSI C++ programming. A developer is recommended to read the following articles:

1. [Symbian OS coding conventions](#)
2. [Symbian OS programming workbook](#)
3. [Symbian OS client and server](#)
4. [Symbian OS Platform Independent Engine Development](#)
5. [S60 Platform Comparison of ANSI C++ and Symbian C++](#)
6. [Symbian OS Creating Custom Controls](#)
7. [Symbian OS view architecture](#)
8. [Symbian OS active object and scheduler](#)
9. [Scalable UI](#)

10. How do i start programming

During development phase, plan the application in advance for localization and Scalable UI since the application will possibly run on all the devices with different screen sizes and orientations. In addition, the application may support multiple languages as well. Defining the localized strings in a **.loc** or **.rls** file is always a good practice to enable good localization support. Do not define the localized strings in the source code **.cpp** files or in **.rss** file if localization support is needed. A good localization example code can be found at:

\\S60\devices\S60_3rd_FP2_SDK\S60CppExamples\Localization

The sample code can only be found in S60 3rd Edition FP2 SDK installation , or

ChineseDisplay sample code. Another good article on localization can also be found at:Symbian OS Multilingual Application Development

when porting an application from previous releases, it is very important to define the stack size in the **.mmp** file. On S60 3rd Edition, it needs a bigger size of stack. Normally, the stack size is defined as follows:

EPOCSTACKSIZE 0x5000

If not defined as above, some unknown errors may happen when the application is running.

Please be noted that the project files should be resided in the same drive as the drive where S60 3rd Edition SDK is installed. Otherwise, error may happen!

2. Building

Use the build function in the IDE tool or use commands in a MS-DOS box to build the example code. For information on how to build an application, see the SDK help.

For Windows emulator, a developer may issues the following command in the **group** directory of the project to build a project once environment is set correctly:

```
bldmake bldfiles
abld build winscw udeb
```

For phone target, the following commands can be issued to build a project:

```
bldmake bldfiles
abld build gcce urel
```

Here, the **GCCE** compiler is free, and it comes with the installation of S60 3rd Edition SDK.

For some of the commercial and serious applications, if performance is of a great concern, a developer is recommended to purchase commercial **RCVT** (ARMV5) compiler. By using the compiler, the target code size is much smaller than the one from **GCCE** compiler. As such, the execution becomes faster, and performance is improved!

In some of the IDE tools (Carbide.c++ and Carbide.vs), the compiler has been seamlessly integrated into the tool, and a user just select the targets, and the targets can be easily built and signed during the compilation process if the certificate information is configured correspondingly.

3. Testing

You can build and run the application in the PC emulator environment. If there are no problems, the application can be deployed into a real target for testing. On S60 3rd Edition, the deployment is a little different from S60 2nd Edition, and Symbian Signed should be involved. During the development phase, you can apply for a developer certificate to sign the developed application for testing. Information on Symbian Signed and how to apply for a developer certificate will be detailed below.

You can find more information about test criteria on Symbian Signed at <https://www.symbiansigned.com/app/page/overview/testcriteria>

If an application needs to be pre-installed onto a Nokia device for some business reasons, it needs to pass the Nokia test criteria. More detailed information can be found at http://www.forum.nokia.com/main/technical_services/testing/index.html

4. Signing

It is compulsory to sign all S60 applications before deploying the application onto a real device. There are two kinds of signing:

- **Self-Signed:** for applications using UIDs between **0x80000000?0xFFFFFFFF**, which belong to the unprotected range, and the application can only use some of the basic set of capabilities. See details on [How to self-sign](#). A developer may generate a self-sign certificate to sign his/her developed application for free. Please be noted that a self-signed application can only make use of the capabilities: **LocalServices**, **ReadUserData**, **WriteUserData**, **NetworkService** and **UserEnvironment** as shown in Figure 4. Of course, a developer may also send the applications with the above capabilities for Symbian signed for some purposes (may be required by the distributors). In this case, a developer needs to change the application UID to protected range for Symbiansigned. A developer may refer to the article [How to sign a sis file with a self-sign certificateto](#) know more about self-signed.
- **Symbian Signed:** for applications using UIDs between **0x00000000?0x7FFFFFFF**, which belong to the protected range, and the application may use some of the sensitive APIs. Regarding Symbiansigned, it is very hard to explain in one word. For more details, please download the webinar: [File:Symbian Signed - Webinar.pdf](#)

Access Capability	User Grantable	Open Signed Online	Open Signed Offline	Express Signed	Certified Signed	Symbian Signed for Nokia
LocalServices ReadUserData WriteUserData NetworkServices UserEnvironment	For testing & sales version	For testing	For testing	Sales version	Sales version	Sales version
Location SwEvent ProtServ TrustedUI PowerMgmt SurroundingsDD ReadDeviceData WriteDeviceData		For testing	For testing	Sales version	Sales version	Sales version
CommDD DiskAdmin MultimediaDD NetworkControl			Device Manufacturer approval			Sales version
AllFiles DRM TCB			Device Manufacturer approval			Sales version
Lead-time	Immediate	Immediate	Immediate	Immediate	1 week	1 week
Note	Developer Tested	Upload SIS	Certify on PC	Developer tested	Test house Tested	Test house Tested

Figure 4: S60 capabilities and symbian signed

For more information on the UID ranges, see <https://www.symbiansigned.com/app/page/dev/uidfaq>. Note that the link is only accessible after a successful login.

As shown in Figure 3, the Symbian capabilities can be categorized into four groups:

1. **User capabilities:** LocalServices, ReadUserData, WriteUserData, NetworkServices, UserEnvironment
2. **System capability set 1:** Location, SwEvent, ProtServ, TrustedUI, PowerMgmt, SurroundingDD, ReadDeviceData, WriteDeviceData
3. **System capability set 2:** CommDD, DiskAdmin, MultimediaDD, NetworkControl
4. **Manufacturer:** AllFiles, DRM, TCB, needs approval from manufacturer of mobile phones

Please be noted that the **Location** capability will be move to **User capabilities** (user grantable) category from **S60 3rd Edition feature pack 2** onwards. During development of an application, if a developer is not so sure what are the needed capabilities, please further refer to the article [How to troubleshoot platform security issues](#). When submitting an application for Symbiansigned, it is required that the application only takes the minimum set of the capabilities needed by the application.

Depending on the capabilities used in an S60 application, there are different ways to sign an application:

1. **User grantable:** The compatibilities used in an application are granted at the installation time, and the application UID has to in the unprotected range (**0x80000000-0xFFFFFFFF**). A developer may use

SignSis command to sign a .sis file. Please be noted that the application can only use: **LocalServices**, **ReadUserData**, **WriteUserData**, **NetworkServices**, **UserEnvironment** (From S60 3rd Edition FP2 onwards, a developer may also use **Location** capability).

2. **Open signed without publisher ID:** A user has to register and log into www.symbiansigned.com to upload an application .sis file. After successful uploading, a developer may immediately get the application signed by the website, and the developer may download it from the website. This is for testing purpose and the signed application can only be installed onto one mobile device. When signing, a developer needs to supply his/her **valid** email address together with the device IMEI (phone serial number) number. The signed application is installable within 36 months, and this method is for development purpose.
3. **Open signed with publisher ID:** A developer has to purchase a publisher ID first from [TC TrustCenter](http://TC.TrustCenter). For Chinese developers, a developer may go to WoSign to purchase a publisher ID. With the publisher ID, a developer may apply for a developer certificate from www.symbiansigned.com. Using the certificate, a developer may sign a .sis file with **SignSis** command. This is for testing purpose and the certificate is limited by the number of IMEI contained in the certificate (maximum 1000 IMEI numbers so far). The certificate is valid for 36 months, and this method is for development purpose.
4. **Express signed:** A developer must have a publisher ID first. The developer has to go to www.symbiansigned.com website to login, and pay 20 USD for such a signing per an application. After uploading the needed files to the website. Immediately, the developer can get the application signed by the website, and the developer may download it from the website. The signed application is for commercial sales.
5. **Certified signed:** A developer must have a publisher ID first. After an application has been fully tested using either **Open signed** method or **self-signed** method (the application must has a UID between **0x00000000-0x7FFFFFFF**), the developer may send the application to one of the 3 named **test houses** in the world for detailed testing, and some money must be paid for such a test although the price may vary from one test house to another. After the application has passed the test criteria from Symbian, the application can be signed for commercial use.
6. **Symbian signed for Nokia:** Any of the applications needed for pre-installation must pass Nokia test criteria, and a developer must pay for such a test. It has stricter criteria than **Certified signed**, and it should be conducted by 3 named **test houses** in the world. The detailed information can be found at: www.symbiansigned.com. For those developers, who need to use manufacturer capabilities, the developer must submit a valid business reason to justify the purpose of using the capabilities. Once the request has been approved, the developer may use **Open signed with publisher ID** for getting the needed certificate for testing and development. This method is for commercial use!

If a developer wants to sign a **Freeware** application, please refer to the article [How to sign a freeware application](#)

After you have fully tested the application, you should send the application to a testing house for further testing if the application needs to be Symbian Signed. Currently, there are three test houses worldwide:

- **mPhasis**
- **Capgmini**
- **NSTL**

The prices for each test house vary, and you are free to choose your test house.

More information about Symbian Signed and testing criteria can be found at <http://www.symbiansigned.com>. A detailed signing process can be found at <https://www.symbiansigned.com/app/page/overview/process>.

Note that you need to apply for an **TrustCenter** publisher ID when the application is sent for Symbian Signed although it may not be needed when applying for a developer certificate during the development phase. Once you have a valid TrustCenter publisher ID, with the ID you can apply for a test certificate, which works for up to 1000 mobile test devices. Developer certificates are not intended for signing applications for distribution. More detailed information regarding developer certificate can be found at https://www.symbiansigned.com/Developer_Certificate_Request_Process_v2.0.pdf.

For Chinese developers, it can be difficult to apply for an TrustCenter publisher ID for some reasons. Based on an agreement with Verisign, a green channel has been opened for developers based in China. More detailed information can be found at https://www.symbiansigned.com/SymbianSigned-General2006_v1_1.pdf.

If a developer wants to develop freeware for the community, a developer may further refer to the link [Freeware](#) to get the developed application signed for free.

5. Transferring and distributing

When the application has been officially signed, the application is ready for distribution. In the following sections, a few links are provided. Nokia NCD (Nokia Content Discover) is also available for use.

Deployment of S60 3rd Edition

When an application has been developed and fully tested in the emulator environment, the application is ready for deployment onto a real device. As described above, it is compulsory to sign every application on S60 3rd Edition before deploying it onto a real device.

- For self-signed applications, you can use **makekeys** one the command line to generate a key and a certificate to sign the application:

```
makekeys -cert -password 12345 -dname "CN=username OU=NOKIA CO=CHINA EM=xyz@abc.com" mykey.key my
```

Alternatively, you can use **Certificate Request** (which is introduced next) application from Symbian to generate a key. A developer may use certificate and key generated using above method to sign an application with user capabilities. Once an application is signed, the .sis file is only valid for 6 months. [Another approach](#) can be used to generate a certificate which can make the .sis file longer than 6 months.

- For Symbian Signed applications, you can use the following steps to generate the needed developer certificate:
 1. **Download** the latest Certificate Request (**DevCertRequest**) application from <http://www.symbiansigned.com>. Install the application.
 2. Start the **Certificate Request** application. Choose a **.csr** name (certificate request). In step **2**, do not simply type in the key files there in the text editor file but rather click the **...** button on the line of **Private Key** if there is no valid private key file yet. As such, a key named key file will be generated during the creation of a request file.
 3. Finish the rest of the steps, and finally, a **.csr** and a **.key** file are generated.

4. Go to <http://www.symbiansigned.com>. Register an account, and log in to the Web site. When you log in, a **My Symbian Signed** tab will become visible. On the page, select **Open Signed->Request**, and upload the generated **.csr** file. If there are no manufacturer capabilities involved in the certificate request, you can download a **.cer** certificate file immediately at the site once the request is uploaded. For certificates containing manufacturer capabilities, please contact Forum Nokia business managers or business groups for help to justify the business reasons before submitting a request.
5. Build the application for **gccce** and **armv5** if you have this compiler. Normally, the **armv5** compiler has much smaller size than **gccce**, which is a free compiler inside the SDK installation. If a developer wants to purchase the product, please refer to the [ARM](#) website for more information. A developer can use the following commands to build the applications for **gccce**:
bldmake bldfile
abld build gccce urel
6. Make use of the **makesis** command to generate a **.sis** file:
makesis myapp_gcce.pkg
In **myapp_gcce.pkg**, it defines where the files are installed.

Once you get both the **.key** and **.cer** file, you can use the following command to sign the application for deployment onto a real device:

```
signsis myapp_gcce.sis myapp_gcce.sisx mycer.cer mykey.key mypassword
```

where **signsis** is the command used to sign the application and **myapp_gcce.sisx** is the output signed file although the extension name does not really matter in this case. You may need to put the correct path inside the command in order to make the command executed successfully. Once an application has been signed, it can be deployed onto a real device for testing via Bluetooth or infrared.

More information on Symbian signed [video clips](#) can be found on the web.

Some IDEs provide some UI to manage the certificates and keys. You can easily choose the build for **gccce** or **armv5** to automatically sign the application during the build phase. Carbide.c++ and Carbide.vs both have this function.

A Symbian signed eLearning courseware can be found at [Symbian signed eLearning](#)

S60 support and useful links

S60 support

You can find all of the latest information regarding S60 on the following Web sites:

<http://www.s60.com>

<http://www.forum.nokia.com>

For Chinese developers, the following Web site provides localized Chinese documents and latest news updates:

<http://www.forum.nokia.com.cn>

On the Forum Nokia Web site, you can download all the documents and sample codes for free. You can find Symbian C++ examples at

http://www.forum.nokia.com/main/resources/technologies/symbian/code_and_examples.html.

You can use the Forum Nokia [Discussion boards](#) to post questions, which other developers may answer. You can also search by using keywords to find out the relevant posts/threads in the discussion areas. For Chinese developers, there is a separate Chinese discussion area available for use. According to the hit-box of Forum Nokia, it is one of the most active discussion areas on the Web site.

In addition to the Web site and discussion boards, Forum Nokia also provides consultancy packages for developers all over the world. This is a fee-based service, and it is open for all developers. For more information, please contact the Forum Nokia local business development managers or technology support and consulting team.

You can get some training services from Forum Nokia as well. Forum Nokia has certified a number of trainers worldwide, and they provide professional training services for developers. More information can be found at the [Training](#) section on the Forum Nokia Web site.

Forum Nokia also provides a professional support service called the **Forum Nokia Pro** program. For full details, follow the Marketplace link in the Developers section of the Nokia Web site at <http://www.nokia.com/developers>.

Useful links

In addition to the links listed above, there a number of other useful resources:

- <http://www.symbian.com>

You can always find a lot of useful and updated information here. A lot of documents, white papers, sample codes, and specifications are available.

- [Symbian coding conventions](#)
- [Symbian OS basics - workbook](#)
- [Essential S60 UI: Get started](#)
- [Developing platform-independent application engines for Symbian OS--module design](#)
- [S60 C++ library](#)

Describes the S60 C++ library and coding conventions

- [Forum Nokia E-Learning courses](#)

This courseware is available on the Forum Nokia Web site for free. It has a number of packages for you to choose from.

- [S60 port from 2nd Edition to 3rd Editin v1.1](#)

Describes how to port an existing S60 2nd Edition to S60 3rd Edition.

- [How do i start programming for Symbian OS](#)

- Nokia developer program

Nokia developer training program

- Nokia certified training centers

Nokia certified training centers worldwide

- Forum Nokia video casts (Chinese)

This is a Chinese Web site, which includes a number of presentations introducing different technology areas.

- <http://opensource.nokia.com/>

This is an open source area for developers. There are a number of open projects.

- <https://www.symbiansigned.com/>

A Web site for signing applications. You can apply for developer certificates and get the testing documents here.

- S60 testing

Forum Nokia page containing information about S60 application testing.

- <http://www.symbianresources.com>

A site with nice tutorials,etc .

- <http://www.newlc.com>

A site for getting supports tutorials, etc.

- <http://www.my-symbian.com>

A good site for selling applications.

- <http://www.handango.com>

A site for selling and marketing Symbian applications.

- <http://www.mapps.com.cn>

A site for selling Chinese applications.

- <http://www.nokia.com/developers/applications>

Find out all of available applications on Nokia NCD (Nokia Content Discover). NCD is a client installed on Nokia devices, and it provides a way for users to find out the needed applications.

- [Differences between S60 2nd and 3rd Edition](#)
- [Platform Security](#)

[S60 3rd Edition preinstalled and preloaded applications](#) Of course, there are a lot of useful links on the Internet, and it is impossible to list them all here.

- [Chinese version of this article](#)
- [S60 3rd Edition application development resources](#)
- [S60 Platform:identification codes](#)
- [Code snippets table for common use cases](#)

Conclusions

S60 is an advanced platform in the mobile device market and it is widely used by mobile device manufacturers. It is an open platform available for third-party developers, and it keeps changing and evolving. S60 3rd Edition incorporates some new features like platform security. The development process on S60 3rd Edition is a little bit different from its predecessors (S60 2nd Edition) due to the introduction of the new platform security model.

S60 devices will be secure and available across a wide 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 impose additional certification requirements on some applications. The introduction of S60 3rd Edition presents developers with both opportunities and challenges. Nokia also offers developers lucrative opportunities to sell mobile content and [Java Verified?](#) applications through Nokia sales channels: Nokia Software Market and [Nokia Catalogs](#). In addition, some applications may be chosen to be embedded in new Nokia devices.