

# Symbian OS

From Wikipedia, the free encyclopedia

*This article is about the historical Symbian OS. For the current, open source Symbian platform descended from Symbian OS and S60, see [Symbian platform](#).*



**This article has multiple issues.** Please help [improve the article](#) or discuss these issues on the [talk page](#).

- It may be **too technical** for a general audience. Please help [make it more accessible](#). Tagged since December 2009.
- It may require general **cleanup** to meet Wikipedia's [quality standards](#). Tagged since December 2009.

**Symbian OS** is an [operating system](#) (OS) designed for [mobile devices](#) and [smartphones](#), with associated [libraries](#), [user interface](#), frameworks and reference implementations of common tools, originally developed by [Symbian Ltd.](#) It was a descendant of [Psion's](#) EPOC and runs exclusively on [ARM processors](#), although an unreleased [x86](#) port existed.

In 2008, the former [Symbian Software Limited](#) was acquired by [Nokia](#) and a new independent non-profit organisation called the [Symbian Foundation](#) was established. **Symbian OS** and its associated user interfaces [S60](#), [UIQ](#) and [MOAP\(S\)](#) were contributed by their owners to the foundation with the objective of creating the [Symbian platform](#) as a royalty-free, open source software. The platform has been designated as the successor to Symbian OS, following the official launch of the Symbian Foundation in April 2009. The [Symbian platform](#) was officially made available as open source code in February 2010.<sup>[3]</sup>

Devices based on Symbian OS account for 46.9% of [smartphone](#) sales, making it the world's most popular mobile operating system.<sup>[4]</sup>

## Contents [hide]

- 1 Design
- 2 Competition
- 3 Structure
- 4 History
  - 4.1 Psion
  - 4.2 EPOC
    - 4.2.1 EPOC16
    - 4.2.2 EPOC32
    - 4.2.3 EPOC OS Releases 1–5
  - 4.3 Symbian OS 6.0 and 6.1
  - 4.4 Symbian OS 7.0 and 7.0s
  - 4.5 Symbian OS 8.0
    - 4.5.1 Symbian OS 8.1
  - 4.6 Symbian OS 9
    - 4.6.1 Symbian OS 9.1 and open source development
    - 4.6.2 Symbian OS 9.2
    - 4.6.3 Symbian OS 9.3
    - 4.6.4 Symbian OS 9.4
    - 4.6.5 Symbian OS 9.5
  - 4.7 Released as Free and Open Source Software
- 5 Devices that use Symbian OS
- 6 Security
  - 6.1 Malware
  - 6.2 Hacking Symbian
- 7 Developing on Symbian OS
- 8 See also
- 9 Notes and references
- 10 Bibliography
- 11 External links

## Design

Symbian features [pre-emptive multitasking](#) and [memory protection](#), like other operating systems (especially those created for use on desktop computers). EPOC's approach to multitasking was inspired by [VMS](#) and is based on asynchronous server-based events.

Symbian OS was created with three systems design principles in mind:

- the integrity and security of user data is paramount,
- user time must not be wasted, and
- all resources are scarce.

To best follow these principles, Symbian uses a [microkernel](#), has a request-and-callback approach to services, and maintains separation between user interface and engine. The OS is optimised for low-power battery-based devices and for ROM-based systems (e.g. features like XIP and re-entrancy in shared libraries). Applications, and the OS itself, follow an object-oriented design: [Model-view-controller](#) (MVC).

Later OS iterations diluted this approach in response to market demands, notably with the introduction of a real-time kernel and a platform security model in versions 8 and 9.

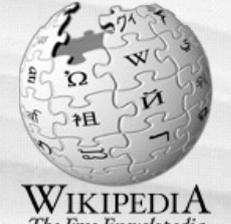
There is a strong emphasis on conserving resources which is exemplified by Symbian-specific programming idioms such as [descriptors](#) and a

## Symbian OS



<b>Company / developer</b>	Nokia/(Symbian Ltd.)
<b>Programmed in</b>	C++ <sup>[1]</sup>
<b>OS family</b>	Mobile operating systems
<b>Working state</b>	development of the original Symbian OS code base has given way for an integrated development of the <a href="#">Symbian platform</a>
<b>Source model</b>	<a href="#">closed source</a> , moved to <a href="#">open source</a> as the <a href="#">Symbian platform</a>
<b>Marketing target</b>	Mobile devices
<b>Supported platforms</b>	ARM, x86 <sup>[2]</sup>
<b>Kernel type</b>	Microkernel
<b>Default user interface</b>	S60 platform, UIQ, MOAP
<b>License</b>	original code base was <a href="#">proprietary</a> , transition to <a href="#">EPL</a> started with Symbian OS 9.1, completed with the <a href="#">Symbian platform</a>
<b>Official website</b>	defunctio <span>­</span> n - see the website of the <a href="#">Symbian Foundation</a>

[\[edit\]](#)



WIKIPEDIA  
The Free Encyclopedia

### navigation

- [Main page](#)
- [Contents](#)
- [Featured content](#)
- [Current events](#)
- [Random article](#)

### search




### interaction

- [About Wikipedia](#)
- [Community portal](#)
- [Recent changes](#)
- [Contact Wikipedia](#)
- [Donate to Wikipedia](#)
- [Help](#)

### toolbox

- [What links here](#)
- [Related changes](#)
- [Upload file](#)
- [Special pages](#)
- [Permanent link](#)
- [Cite this page](#)

### print/export

- [Create a book](#)
- [Download as PDF](#)
- [Printable version](#)

### languages

- العربية
- [Azərbaycan](#)
- Български
- [Català](#)
- [Česky](#)
- [Dansk](#)
- [Deutsch](#)
- [Ελληνικά](#)
- [Español](#)
- فارسی
- [Français](#)
- [Galego](#)
- [한국어](#)
- [Bahasa Indonesia](#)
- [Italiano](#)
- [עברית](#)
- [ქართული](#)
- [Kurdî](#)
- [Latviešu](#)
- [Lietuvių](#)
- [Magyar](#)
- [Nederlands](#)
- [日本語](#)
- [Norsk \(bokmål\)](#)
- [Norsk \(nynorsk\)](#)
- [O'zbek](#)
- [Polski](#)
- [Português](#)
- [Română](#)
- [Русский](#)
- [Slovenčina](#)
- [Slovenščina](#)
- [Sorani / کوردی](#)
- [Српски / Srpski](#)
- [Suomi](#)
- [Svenska](#)
- [ไทย](#)
- [Türkçe](#)
- [Українська](#)
- [Tiếng Việt](#)
- [中文](#)

**cleanup stack**. There are similar techniques for conserving disk space (though the disks on Symbian devices are usually **flash memory**). Furthermore, all Symbian programming is event-based, and the **CPU** is switched into a low power mode when applications are not directly dealing with an event. This is achieved through a programming idiom called **active objects**. Similarly the Symbian approach to threads and processes is driven by reducing overheads.

The Symbian kernel (**EKA2**) supports sufficiently-fast **real-time** response to build a single-core phone around it — that is, a phone in which a single processor core executes both the user **applications** and the **signalling stack**<sup>[5]</sup>. This has allowed Symbian EKA2 phones to become smaller, cheaper and more power efficient than their predecessors<sup>[citation needed]</sup>.

## Competition

[edit]

In the number of "smart mobile device" shipments, Symbian devices are the market leaders. Statistics published in February 2010 showed that the Symbian devices comprised a 47.2% share of the smart mobile devices shipped in in 2009, with **RIM** having 20.8%, Apple having 15.1% (through **iPhone OS**), Microsoft having 8.8% (through **Windows CE** and **Windows Mobile**) and **Android** having 4.7%.<sup>[6]</sup> Other competitors include **Palm OS**, **Qualcomm's BREW**, **SavaJe**, **Linux** and **MontaVista Software**.

Although the share of the global smartphone market dropped from 52.4% in 2008 to 47.2% in 2009, the shipment volume of Symbian devices grew 4.8%, from 74.9 million units to 78.5 million units.<sup>[6]</sup>

## Structure

[edit]

The Symbian System Model contains the following layers, from top to bottom:

- UI **Framework** Layer
- Application Services Layer
  - **Java ME**
- OS Services Layer
  - generic OS services
  - communications services
  - multimedia and graphics services
  - connectivity services
- Base Services Layer
- Kernel Services & Hardware Interface Layer

The Base Services Layer is the lowest level reachable by user-side operations; it includes the **File Server** and User Library, a Plug-In **Framework** which manages all plug-ins, Store, Central Repository, **DBMS** and cryptographic services. It also includes the Text Window Server and the Text Shell: the two basic services from which a completely functional port can be created without the need for any higher layer services.

Symbian has a **microkernel** architecture, which means that the minimum necessary is within the kernel to maximise robustness, availability and responsiveness. It contains a **scheduler**, **memory management** and device drivers, but other services like networking, telephony and **filesystem** support are placed in the OS Services Layer or the Base Services Layer. The inclusion of device drivers means the kernel is not a *true* microkernel. The **EKA2** real-time kernel, which has been termed a **nanokernel**, contains only the most basic primitives and requires an extended kernel to implement any other abstractions.

Symbian is designed to emphasise compatibility with other devices, especially removable media file systems. Early development of EPOC led to adopting **FAT** as the internal file system, and this remains, but an object-oriented persistence model was placed over the underlying FAT to provide a **POSIX**-style interface and a streaming model. The internal data formats rely on using the same APIs that create the data to run all file manipulations. This has resulted in data-dependence and associated difficulties with changes and data migration.

There is a large networking and communication subsystem, which has three main servers called: ETEL (EPOC telephony), ESOCK (EPOC sockets) and C32 (responsible for serial communication). Each of these has a plug-in scheme. For example ESOCK allows different ".PRT" protocol modules to implement various networking protocol schemes. The subsystem also contains code that supports short-range communication links, such as **Bluetooth**, **IrDA** and **USB**.

There is also a large volume of **user interface** (UI) Code. Only the base classes and substructure were contained in Symbian OS, while most of the actual user interfaces were maintained by third parties. This is no longer the case. The three major UIs - S60, UIQ and MOAP - were contributed to Symbian in 2009. Symbian also contains graphics, text layout and font rendering libraries.

All native Symbian C++ applications are built up from three framework classes defined by the application architecture: an application class, a document class and an application user interface class. These classes create the fundamental application behaviour. The remaining required functions, the application view, data model and data interface, are created independently and interact solely through their APIs with the other classes.

Many other things do not yet fit into this model – for example, **SyncML**, **Java ME** providing another set of APIs on top of most of the OS and **multimedia**. Many of these are frameworks, and vendors are expected to supply plug-ins to these frameworks from third parties (for example, **Helix Player** for multimedia **codecs**). This has the advantage that the APIs to such areas of functionality are the same on many phone models, and that vendors get a lot of flexibility. But it means that phone vendors needed to do a great deal of integration work to make a Symbian OS phone.

Symbian includes a reference user-interface called "TechView". It provides a basis for starting customisation and is the environment in which much Symbian test and example code runs. It is very similar to the user interface from the **Psion Series 5 personal organiser** and is not used for any production phone user interface.

## History

[edit]



It has been suggested that this section be **split** into a new article. ([Discuss](#))

### Psion

[edit]

In 1980, **Psion** was founded by **David Potter**.

### EPOC

[edit]

**EPOC** is a family of **graphical operating systems** developed by **Psion** for portable devices, primarily **PDA**s. EPOC came from **epoch**, the beginning of an era, but was backfitted by the engineers to



"Electronic Piece Of Cheese".<sup>[7]</sup>

## EPOC16

[[edit](#)]

EPOC16, originally simply named EPOC, was the operating system developed by Psion in the late 1980s and early 1990s for Psion's "SIBO" (Sixteen Bit Organisers) devices. All EPOC16 devices featured an **8086**-family processor and a **16-bit** architecture. EPOC16 was a single-user **pre-emptive multitasking** operating system, written in Intel 8086 **assembler language** and **C** and designed to be delivered in **ROM**. It supported a simple programming language called **Open Programming Language** (OPL) and an **integrated development environment** (IDE) called OVAL. SIBO devices included the: MC200, MC400, **Series 3** (1991–1998), Series 3a, Series 3c, Series 3mx, Siena, Workabout and Workabout mx. The MC400 and MC200, the first EPOC16 devices, shipped in 1989.



Osaris PDA by Oregon Scientific with EPOC operating system

EPOC16 featured a primarily 1-bit-per-pixel, keyboard-operated **graphical interface**  (the hardware for which it was designed did not have **pointer** input).

In the late 1990s, the operating system was referred to as **EPOC16** to distinguish it from Psion's then-new EPOC32 OS.

## EPOC32

[[edit](#)]

The first version of EPOC32, Release 1 appeared on the Psion **Series 5** ROM v1.0 in 1997. Later, ROM v1.1 featured Release 3 (Release 2 was never publicly available.) These were followed by the **Psion Series 5mx**, **Revo** / Revo plus, **Psion Series 7** / **netBook** and netPad (which all featured Release 5).

The EPOC32 operating system, at the time simply referred to as EPOC, was later renamed Symbian OS. Adding to the confusion with names, before the change to Symbian, EPOC16 was often referred to as SIBO to distinguish it from the "new" EPOC. Despite the similarity of the names, EPOC32 and EPOC16 were completely different operating systems, EPOC32 being written in C++ from a new codebase with development beginning during the mid 1990s.

EPOC32 was a **pre-emptive multitasking**, single user operating system with memory protection, which encourages the application developer to separate their program into an engine and an **interface**. The Psion line of PDAs come with a **graphical user interface** called **EIKON** which is specifically tailored for handheld machines with a keyboard (thus looking perhaps more similar to desktop GUIs than palmtop GUIs [2] ). However, one of EPOC's characteristics is the ease with which new GUIs can be developed based on a core set of GUI classes, a feature which has been widely explored from **Ericsson R380** and onwards.

EPOC32 was originally developed for the **ARM** family of processors, including the **ARM7**, **ARM9**, **StrongARM** and Intel's **XScale**, but can be compiled towards target devices using several other processor types.

During the development of EPOC32, Psion planned to license EPOC to third-party device manufacturers, and spin off its software division as Psion Software. One of the first licensees was the short-lived **Geofax**, which halted production with less than 1,000 units sold. **Ericsson** marketed a rebranded Psion Series 5mx called the **MC218**, and later created the EPOC Release 5.1 based **smartphone**, the **R380**. **Oregon Scientific** also released a budget EPOC device, the **Osaris** (notable as the only EPOC device to ship with Release 4).

In June 1998, Psion Software became **Symbian Ltd.**, a major joint venture between Psion and phone manufacturers **Ericsson**, **Motorola**, and **Nokia**. As of Release 6, EPOC became known simply as Symbian OS.

## EPOC OS Releases 1–5

[[edit](#)]

Work started on the 32-bit version in late 1994.

The **Series 5** device, released in June 1997, used the first iterations of the EPOC32 OS, codenamed "Protea", and the "Eikon" graphical user interface.

The Oregon Scientific **Osaris** was the only PDA to use the ER4.

The **Psion Series 5mx**, **Psion Series 7**, **Psion Revo**, **Diamond Mako**, **Psion netBook** and **Ericsson MC218** were released in 1999 using ER5. A phone project was announced at **CeBIT**, the Phillips Illium/Accent, but did not achieve a commercial release. This release has been retrospectively dubbed Symbian OS 5.

The first phone using ER5u, the **Ericsson R380** was released in November 2000. It was not an 'open' phone – software could not be installed. Notably, a number of never-released Psion prototypes for next generation PDAs, including a Bluetooth Revo successor codenamed **Conan** were using ER5u. The 'u' in the name refers to the fact that it supported **Unicode**.

## Symbian OS 6.0 and 6.1

[[edit](#)]

The OS was renamed Symbian OS and was envisioned as the base for a new range of **smartphones**. This release sometime is called ER6. Psion gave 130 key staff to the new company and retained a 31% shareholding in the spun-out business.

The first 'open' Symbian OS phone, the **Nokia 9210** Communicator, was released in June 2001. **Bluetooth** support was added. Almost 500,000 Symbian phones were shipped in 2001, rising to 2.1 million the following year.

Development of different UIs was made generic with a "reference design strategy" for either 'smartphone' or 'communicator' devices, subdivided further into keyboard- or tablet-based designs. Two reference UIs (DFRDs or Device Family Reference Designs) were shipped – Quartz and Crystal. The former was merged with Ericsson's 'Ronneby' design and became the basis for the **UIQ** interface; the latter reached the market as the Nokia **Series 80** UI.

Later DFRDs were Sapphire, Ruby, and Emerald. Only Sapphire came to market, evolving into the Pearl DFRD and finally the Nokia **Series 60** UI, a keypad-based 'square' UI for the first true smartphones. The first one of them was the **Nokia 7650** smartphone (featuring Symbian OS 6.1), which was also the first with a built-in camera, with VGA (0.3 Mpx = 640×480) resolution.

Despite these efforts to be generic, the UI was clearly split between competing companies: Crystal or Sapphire was Nokia, Quartz was Ericsson. DFRD was abandoned by Symbian in late 2002, as part of an active retreat from UI development in favour of 'headless' delivery. Pearl was given to Nokia, Quartz development was spun off as UIQ Technology AB, and work with Japanese firms was quickly folded into the **MOAP** standard.

## Symbian OS 7.0 and 7.0s

[[edit](#)]

First shipped in 2003. This is an important Symbian release which appeared with all contemporary user interfaces including **UIQ** (Sony Ericsson P800, P900, P910, Motorola A925, A1000), **Series 80** (Nokia 9300, 9500), **Series 90** (Nokia 7710), **Series 60** (Nokia 3230, 6260, 6600, 6670, 7610) as well as several **FOMA** phones in Japan. It also added **EDGE** support and **IPv6**. Java support was changed from pJava and JavaPhone to one based on the Java ME standard.

One million Symbian phones were shipped in Q1 2003, with the rate increasing to one million a month by the end of 2003.

Symbian OS 7.0s was a version of 7.0 special adapted to have greater [backward compatibility](#) with Symbian OS 6.x, partly for compatibility between the Communicator 9500 and its predecessor the Communicator 9210.

In 2004, Psion sold its stake in Symbian. The same year, the first [worm](#) for mobile phones using Symbian OS, *Cabir*, was developed, which used [Bluetooth](#) to spread itself to nearby phones. See [Cabir](#) and [Symbian OS threats](#).

## Symbian OS 8.0

[[edit](#)]

First shipped in 2004, one of its advantages would have been a choice of two different kernels ([EKA1](#) or [EKA2](#)). However, the EKA2 kernel version did not ship until Symbian OS 8.1b. The kernels behave more or less identically from user-side, but are internally very different. EKA1 was chosen by some manufacturers to maintain compatibility with old device drivers, while EKA2 was a [real-time](#) kernel. 8.0b was deproductised in 2003.

Also included were new APIs to support [CDMA](#), [3G](#), two-way data streaming, [DVB-H](#), and [OpenGL ES](#) with [vector graphics](#) and direct screen access.

## Symbian OS 8.1

[[edit](#)]

An improved version of 8.0, this was available in 8.1a and 8.1b versions, with EKA1 and EKA2 kernels respectively. The 8.1b version, with EKA2's single-chip phone support but no additional security layer, was popular among Japanese phone companies desiring the real-time support but not allowing open application installation. The first and maybe the most famous smartphone featuring Symbian OS 8.1a was [Nokia N90](#) in 2005, [Nokia](#)'s first in [Nseries](#).

## Symbian OS 9

[[edit](#)]

Symbian OS 9.0 was used for internal Symbian purposes only. It was de-productised in 2004. 9.0 marked the end of the road for EKA1. 8.1a is the final EKA1 version of Symbian OS.

Symbian OS has generally maintained reasonable [binary code compatibility](#). In theory the OS was BC from ER1-ER5, then from 6.0 to 8.1b. Substantial changes were needed for 9.0, related to tools and security, but this should be a one-off event. The move from requiring ARMv4 to requiring ARMv5 did not break backwards compatibility.

## Symbian OS 9.1 and open source development

[[edit](#)]

Released early 2005. It includes many new security related features, including platform security module facilitating [mandatory code signing](#). The new ARM [EABI](#) binary model means developers need to retool and the security changes mean they may have to recompile. [S60 platform](#) 3rd Edition phones have Symbian OS 9.1. Sony Ericsson is shipping the [M600](#) and [P990](#) based on Symbian OS 9.1. The earlier versions had a defect where the phone hangs temporarily after the owner sent a large number of SMS'es. However, on 13 September 2006, Nokia released a small program to fix this defect.<sup>[8]</sup> Support for [Bluetooth](#) 2.0 was also added.

Symbian 9.1 introduced [capabilities](#) and a [Platform Security](#) framework. To access certain APIs, developers have to sign their application with a [digital signature](#). Basic capabilities are user-grantable and developers can [self-sign](#) them, while more advanced capabilities require certification and signing via the [Symbian Signed](#) program, which uses independent 'test houses' and phone manufacturers for approval. For example, file writing is a user-grantable capability while access to Multimedia Device Drivers require phone manufacturer approval. A TC TrustCenter [ACS Publisher ID certificate](#) is required by the developer for signing applications.

## Symbian OS 9.2

[[edit](#)]

Released Q1 2006. Support for [OMA Device Management](#) 1.2 (was 1.1.2). Vietnamese language support. [S60](#) 3rd Edition Feature Pack 1 phones have Symbian OS 9.2. Nokia phones with Symbian OS 9.2 OS include the [Nokia E90](#), [Nokia N95](#), [Nokia N82](#) and [Nokia 5700](#).

## Symbian OS 9.3

[[edit](#)]

Released on 12 July 2006. Upgrades include improved memory management and native support for Wifi [802.11](#), [HSDPA](#). The [Nokia E72](#), [Nokia 5730 XpressMusic](#), [Nokia N79](#), [Nokia N96](#), [Nokia E52](#), [Nokia E75](#) and others feature Symbian OS 9.3.

## Symbian OS 9.4

[[edit](#)]

Announced in March 2007. Provides the concept of demand paging which is available from v9.3 onwards. Applications should launch up to 75% faster. Additionally, [SQL](#) support is provided by [SQLite](#). Ships with the [Samsung i8910 Omnia HD](#), [Nokia N97](#), [Nokia 5800 XpressMusic](#), [Nokia 5530 XpressMusic](#) and [Sony Ericsson Satio](#). Used as the basis for Symbian^1, the first [Symbian platform](#) release. The release is also better known as [S60 5th edition](#), as it is the bundled interface for the OS.

## Symbian OS 9.5

[[edit](#)]

In 26 March 2007 [Symbian Ltd.](#) announced v9.5 which includes native support for mobile [digital television](#) broadcasts in [DVB-H](#) and [ISDB-T](#) formats and also location services.<sup>[9]</sup>

## Released as Free and Open Source Software

[[edit](#)]

The [Symbian Foundation](#) was announced in June 2008 and came into existence in 2009. Its objective was to publish the source for the entire [Symbian platform](#) under the [OSI](#)- and [FSF](#)-approved [Eclipse Public License \(EPL\)](#). The release of the [Symbian platform](#) deprecated Symbian OS as a standalone product.

## Devices that use Symbian OS

[[edit](#)]

On 16 November 2006, the 100 millionth [smartphone](#) running the OS was shipped.<sup>[10]</sup>

- The [Ericsson R380](#), in 2000, was the first commercially available phone based on Symbian OS. As with the modern "FOMA" phones, this device was closed, and the user could not install new C++ applications. Unlike those, however, the R380 could not even run Java applications, and for this reason, some have questioned whether it can properly be termed a 'smartphone'.
- The [UIQ](#) interface was used for PDAs such as [Sony Ericsson P800](#), [P900](#), [W950](#) and the [RIZR Z8](#) and [RIZR Z10](#).
- The Nokia [S60](#) interface is used in various phones, the first being the [Nokia 7650](#). The [Nokia N-Gage](#) and [Nokia N-Gage QD](#) gaming/smartphone combos are also S60 platform devices. It was also used on other manufacturers' phones such as the [Siemens SX1](#) and [Samsung SGH-Z600](#). Recently, more advanced devices using S60 include the Nokia 6xxx, the [Nseries](#) (except Nokia N8xx and N9xx), the [Eseries](#) and some models of the Nokia [XpressMusic](#) mobiles.
- The [Nokia 9210](#), [9300](#) and [9500](#) Communicator smartphones used the [Nokia Series 80](#) interface.

- The **Nokia 7710** is the only device currently using the **Nokia Series 90** interface.
- **Fujitsu**, **Mitsubishi**, **Sony Ericsson** and **Sharp** developed phones for **NTT DoCoMo** in Japan, using an interface developed specifically for DoCoMo's **FOMA** "Freedom of Mobile Access" network brand. This UI platform is called **MOAP** "Mobile Oriented Applications Platform" and is based on the UI from earlier Fujitsu FOMA models.

As of 21 July 2009, more than 250 million devices running Symbian OS had been shipped.<sup>[11]</sup>

## Security

[edit]

### Malware

[edit]

*Main article: [Mobile virus](#)*

Symbian OS was subject to a variety of viruses, the best known of which is **Cabir**. Usually these send themselves from phone to phone by Bluetooth. So far, none have taken advantage of any flaws in Symbian OS – instead, they have all asked the user whether they would like to install the software, with somewhat prominent warnings that it can't be trusted.

However, with a view that the average mobile phone user shouldn't have to worry about security, Symbian OS 9.x adopted a UNIX-style **capability** model (permissions per process, not per object). Installed software is theoretically unable to do damaging things (such as costing the user money by sending network data) without being digitally signed – thus making it traceable. Commercial developers who can afford the cost can apply to have their software signed via the **Symbian Signed**  program. Developers also have the option of self-signing their programs. However, the set of available features does not include access to Bluetooth, IrDA, GSM CellID, voice calls, GPS and few others. Some operators have opted to disable all certificates other than the Symbian Signed certificates.

Some other hostile programs are listed below, but all of them still require the input of the user to run.

- **Drever.A** is a malicious SIS file trojan that attempts to disable the automatic startup from Simworks and Kaspersky Symbian Anti-Virus applications.
- **Locknut.B** is a malicious SIS file trojan that pretends to be patch for Symbian S60 mobile phones. When installed, it drops a binary that will crash a critical system service component. This will prevent any application from being launched in the phone.
- **Mabir.A** is basically **Cabir** with added MMS functionality. The two are written by the same author, and the code shares many similarities. It spreads using **Bluetooth** via the same routine as early variants of Cabir. As Mabir. A activates it will search for the first phone it finds, and starts sending copies of itself to that phone.
- **Fontal.A** is an SIS file trojan that installs a corrupted file which causes the phone to fail at reboot. If the user tries to reboot the infected phone, it will be permanently stuck on the reboot, and cannot be used without disinfection – that is, the use of the reformat key combination which causes the phone to lose all data. Being a trojan, Frontal cannot spread by itself – the most likely way for the user to get infected would be to acquire the file from untrusted sources, and then install it to the phone, inadvertently or otherwise.

### Hacking Symbian

[edit]

S60 v3 and v5 (OS 9.x) devices can be hacked to remove the platform security introduced in OS 9.1 onwards thus allowing users to install "unsigned" files (files without certificates validated by Symbian) and allowing access to previously locked system files.<sup>[12]</sup> This allows changing of how the operating system works, allowing hidden applications etc. to be viewable and possibly increases the threat posed by mobile viruses as the operating system files are now exposed.<sup>[13]</sup>

## Developing on Symbian OS

[edit]

The native language of Symbian is C++, although it is not a standard implementation. There were multiple platforms based upon Symbian OS that provided **SDKs** for application developers wishing to target Symbian OS devices – the main ones being UIQ and S60. Individual phone products, or families, often had SDKs or SDK extensions downloadable from the manufacturer's website too. With the various UI platforms unified in the Symbian platform there should be less diversity between manufacturer's SDKs from 2010 onwards.

The SDKs contain documentation, the header files and library files required to build Symbian OS software, and a Windows-based emulator ("WINS"). Up until Symbian OS version 8, the SDKs also included a version of the **GCC** compiler (a **cross-compiler**) required to build software to work on the device.

Symbian OS 9 and the Symbian platform use a new **ABI** and require a different compiler – a choice of compilers is available including a newer version of GCC (see external links below).

Unfortunately, Symbian C++ programming has a steep **learning curve**, as Symbian requires the use of special techniques such as descriptors and the cleanup stack. This can make even relatively simple programs harder to implement than in other environments. Moreover, it is questionable whether Symbian's techniques, such as the memory management paradigm, are actually beneficial. It is possible that the techniques, developed for the much more restricted mobile hardware of the 1990s, simply cause unnecessary complexity in source code because programmers are required to concentrate on low-level routines instead of more application-specific features. It seems difficult, however, to make a move towards a more high-level and modern programming paradigm.<sup>[14]</sup>

Symbian C++ programming is commonly done with an **IDE**. For previous versions of Symbian OS, the commercial IDE **CodeWarrior** for Symbian OS was favoured. The CodeWarrior tools were replaced during 2006 by **Carbide.c++**, an **Eclipse**-based IDE developed by Nokia. Carbide.c++ is offered in four different versions: Express, Developer, Professional, and OEM, with increasing levels of capability. Fully featured software can be created and released with the Express edition, which is free. Features such as UI design, crash debugging etc. are available in the other, charged-for, editions. **Microsoft Visual Studio** 2003 and 2005 are also supported through the **Carbide.vs** plugin.

Symbian's flavour of **C++** is very specialised.<sup>[citation needed]</sup> However, Symbian devices can also be programmed using **Python**, **Java ME**, **Flash Lite**, **Ruby**, **.NET**, **Web Runtime (WRT)** Widgets and Standard C/C++.<sup>[15]</sup>

Visual Basic programmers can use **NS Basic** to develop apps for S60 3rd Edition and UIQ 3 devices.

In the past, Visual Basic, VB.NET, and C# development for Symbian were possible through **AppForge** Crossfire, a plugin for Microsoft Visual Studio. On 13 March 2007 AppForge ceased operations; Oracle purchased the intellectual property, but **announced**  that they did not plan to sell or provide support for former AppForge products. **Net60** , a .NET compact framework for Symbian, which is developed by redFIVElabs, is sold as a commercial product. With Net60, VB.NET and C# (and other) source code is compiled into an intermediate language (IL) which is executed within the Symbian OS using a just-in-time compiler. (As of 18/1/10 RedFiveLabs has ceased development of Net60 with this announcement on their landing page: At this stage we are pursuing some options to sell the IP so that Net60 may continue to have a future.)

There is also a version of a **Borland** IDE for Symbian OS. Symbian OS development is also possible on **Linux** and **Mac OS X** using tools and techniques developed by the community, partly enabled by Symbian releasing the source code for key tools. A plugin that allows development of

Symbian OS applications in Apple's [Xcode](#) IDE for Mac OS X is available.<sup>[16]</sup>

Once developed, Symbian applications need to find a route to customers' mobile phones. They are packaged in SIS files which may be installed over-the-air, via PC connect, Bluetooth or on a memory card. An alternative is to partner with a phone manufacturer and have the software included on the phone itself. The SIS file route is more difficult for Symbian OS 9.x, because any application wishing to have any capabilities beyond the bare minimum must be signed via the [Symbian Signed](#) program. There are however various hacks, which allow installing unsigned programs with any capabilities to Symbian OS 9.x.

Introduction of the [Symbian Signed](#) system in which application developers need to pay in order to use some of the more attractive smartphone features (as contrasted to platforms like [Palm OS](#) and [Windows Mobile](#)) is making it an increasingly unpopular platform for Open Source projects<sup>[17]</sup>, independent developers and small startups. This situation is worsened by the fragmentation of user interface systems (UIQ vs S60 vs MOAP)<sup>[18]</sup>, meaning that developers need to build and maintain multiple incompatible versions of their software<sup>[19]</sup> if they want to target multiple devices which use the same underlying Symbian OS version.

[Java ME](#) applications for Symbian OS are developed using standard techniques and tools such as the [Sun Java Wireless Toolkit](#) (formerly the J2ME Wireless Toolkit). They are packaged as JAR (and possibly JAD) files. Both CLDC and CDC applications can be created with [NetBeans](#). Other tools include [SuperWaba](#), which can be used to build Symbian 7.0 and 7.0s programs using Java.

Nokia S60i phones can also run [Python](#) scripts when the interpreter [Python for S60](#) is installed, with a custom made API that allows for Bluetooth support and such. There is also an interactive console to allow the user to write python scripts directly from the phone.

## See also

[edit]

- [Symbian platform](#), the continuation of the Symbian OS by the [Symbian Foundation](#)
- [Accredited Symbian Developer](#)
- [MOAP\(S\)](#)
- [UIQ](#)
- [Series 60](#)
- [Python for S60](#)
- [Mobile Web Server](#)
- [P.I.P.S. Is POSIX on Symbian](#)
- [Maemo](#)
- [MeeGo](#)

## Notes and references

[edit]

- ↑ Lextrait, Vincent (January 2010). "The Programming Languages Beacon, v10.0". Retrieved 5 January 2010.
  - ↑ symbian on intel's atom architecture
  - ↑ http://www.watblog.com/2010/02/06/symbian-os-now-fully-open-source/
  - ↑ http://www.gartner.com/it/page.jsp?id=1306513
  - ↑ Introducing EKA2, by Jane Sales with Martin Tasker
  - ↑ ***a b*** [1]
  - ↑ http://archive.salon.com/tech/view/2000/05/15/colly\_myers/index.html
  - ↑ Solution to Nokia Slow SMS / Hang Problem
  - ↑ http://www.allaboutsymbian.com/news/item/5075\_Symbian\_OS\_95\_.php
  - ↑ Six Years of Symbian Produces 100 Models and 100 Million Shipments, The Smart PDA.
  - ↑ Symbian Foundation Adds New Member, Nuance
  - ↑ Nokia's S60 3rd Ed security has been HACKED? - Symbian Freak
  - ↑ Symbian Freak: S60 v3 Hacking – Mission accomplished, FP1 hacked!!
  - ↑ Symbian OS design faults
  - ↑ http://developer.symbian.org
  - ↑ Tom Sutcliffe and Jason Barrie Morley Xcode Symbian support
  - ↑ Why Symbian Signed must die
  - ↑ List of Symbian phones with platforms
  - ↑ List of SDKs for specific platforms
20. [www.symbianism.com](#)

## Bibliography

[edit]

- Morris, Ben (22 June 2007), *The Symbian OS architecture sourcebook : design and evolution of a mobile phone OS*, John Wiley & Sons, pp. 630, ISBN 0470018461

## External links

[edit]

- [Symbian Foundation home page](#)
- [Symbian Developer website](#) – the official developer network
- [Symbian OS](#) at the [Open Directory Project](#)

<span>v</span> <span>.</span> <span>d</span> <span>.</span> <span>e</span>	<b>Nokia platforms</b>	[show]
<span>v</span> <span>.</span> <span>d</span> <span>.</span> <span>e</span>	<b>Mobile operating systems</b>	[show]
<span>v</span> <span>.</span> <span>d</span> <span>.</span> <span>e</span>	<b>Real-time operating systems (RTOS)</b>	[show]

Categories: [Smartphones](#) | [Personal digital assistants](#) | [Real-time operating systems](#) | [Embedded operating systems](#) | [Microkernel-based operating systems](#) | [Mobile phone operating systems](#) | [Mobile operating systems](#) | [Mobile open source](#) | [Symbian OS](#) | [Nokia](#)

