

---

# **S60 Platform: Identification Codes**

**Version 2.0**  
November 2, 2007

**S60** platform

## Legal notice

Copyright © 2002–2007 Nokia Corporation. All rights reserved.

Nokia and Forum Nokia are registered trademarks of Nokia Corporation. Other product and company names mentioned herein may be trademarks or trade names of their respective owners.

### Disclaimer

The information in this document is provided “as is,” with no warranties whatsoever, including any warranty of merchantability, fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification, or sample. Furthermore, information provided in this document is preliminary, and may be changed substantially prior to final release. This document is provided for informational purposes only.

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

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

### License

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

## Contents

<b>1.</b>	<b>Introduction .....</b>	<b>5</b>
<b>2.</b>	<b>Identification codes.....</b>	<b>6</b>
<b>3.</b>	<b>Syntax of defining identification codes .....</b>	<b>7</b>
3.1	Syntax in S60 3rd Edition and later.....	7
3.2	Syntax in S60 1st and 2nd Editions .....	7
<b>4.</b>	<b>Considerations for using Platform IDs and Product IDs .....</b>	<b>8</b>
4.1	Applications supporting multiple platform releases.....	8
4.1.1	S60 3rd Edition and later .....	8
4.1.2	S60 1st and 2nd Editions.....	8
4.2	Device-specific applications.....	8
4.2.1	Devices based on S60 3rd Edition and later .....	8
4.2.2	Devices based on S60 1st or 2nd Edition .....	9
<b>5.</b>	<b>Detecting the device at run time with the Machine UID .....</b>	<b>10</b>
<b>6.</b>	<b>Detecting the platform version during installation .....</b>	<b>11</b>
<b>7.</b>	<b>Detecting the platform version at run time .....</b>	<b>13</b>
<b>Appendix A.</b>	<b>Supported IDs in S60 devices .....</b>	<b>15</b>
	<b>Evaluate this resource.....</b>	<b>18</b>

## Change history

September 18, 2002	Version 1.0	Document added into Forum Nokia
September 26, 2002	Version 1.1	Correction to Product ID number and some instructions added
December 8, 2003	Version 1.2	Updates to include S60 2nd Edition
July 8, 2004	Version 1.3	Machine UIDs added
May 4, 2006	Version 1.4	Major update with structural changes. Updated to include S60 2nd and 3rd Edition IDs.
November 6, 2006	Version 1.5	Methods to detect platform version during installation and at run time added. Appendix A updated to include S60 3rd Edition, Feature Pack 1 ID, and IDs of latest S60 devices.
December 8, 2006	Version 1.6	Product ID updates: Nokia 6290 and Nokia E62 added and Nokia N95 corrected.
January 12, 2007	Version 1.7	Product ID updates: Nokia N76 and Nokia N93i added.
February 22, 2007	Version 1.8	Product ID updates: Nokia 6110 Navigator, Nokia E61i, Nokia E65, Nokia E90, and Nokia N77 added.
April 24, 2007	Version 1.9	Product ID updates: Nokia 5700 XpressMusic and Nokia 6120 Classic added.
November 2, 2007	Version 2.0	Information on checking platform version with the Version Info API added.  Product ID updates: Nokia 6121 Classic, Nokia E51, Nokia N81, and Nokia N95 8GB.

---

## 1. Introduction

Devices based on the S60 platform have a built-in mechanism to warn users attempting to install incompatible software — either a non-S60-based application or an S60-based application targeted at a newer platform release than the release of the device into which the application is being installed. Incompatibility can also be an issue when S60-based applications that are built only for a particular S60 device model are being installed on a different S60 device model. An example of an application that is incompatible with a device model is a camera application that is being installed into a device that does not have a camera.

This document describes how application developers should handle the compatibility of C++ applications at installation time, using S60 identification codes (Platform IDs and Product IDs). Usage of the identification codes is also illustrated. A list of currently supported IDs of S60 devices is included in Appendix A.

In addition, separate concepts for detecting the device identity at run time (using the Machine UID) and detecting the platform version during installation and at run time are briefly described.

---

## 2. Identification codes

An identification code is a special identifier sequence (**id-sequence**) that can be used to identify a platform release or to identify a specific device model, known as a Platform ID or a Product ID, respectively.

The Platform ID is defined for each S60 platform release (Editions and Feature Packs). Thus, if a new feature is supported by a certain S60 platform release and onwards, the Platform ID of that specific release can be used to ensure the installation of an application into compatible devices, provided that there is no binary break between platform releases.

The Product ID is defined for each S60 device. Similarly to the Platform ID, the Product ID prevents installation of an application into a device where it would not run optimally or at all. A typical use case would be, for example, that an application uses a specific software or hardware feature supported by one or a few devices only, and the application would not work in a desired way or at all in any other device. Thus, it would be necessary to make sure that the application could not be installed into incompatible devices. However, the use cases for a Product ID should be rare for devices' software feature dependence, since S60 C++ APIs are commonly targeted at multiple platform versions.

The Platform IDs of each S60 platform release and the product-specific IDs for each S60 device are listed in Table 1 and Table 2.

All installation packages of S60 applications should contain the id-sequence in order to facilitate smooth installation of the software. If the id-sequence is not found or the id-sequence is not recognized by a device (for example, the id-sequence refers to a newer platform release than the release supported by the device), the user will get a notification about the potential incompatibility. Depending on the platform release, the installation process can be continued, but at the risk of application functionality failure.

### 3. Syntax of defining identification codes

The identification code is specified as a requisite in the installation package file (.pkg). The syntax of defining identification codes is slightly different between S60 3rd Edition and S60 1st/2nd Editions.

#### 3.1 Syntax in S60 3rd Edition and later

The identifier sequence line is added right after the line of the installation package header, with the ID embraced by square brackets:

```
#{"MyApplication"}, (0x20000001), 1, 0, 0
[0x101F7961], 0, 0, 0, {"Series60ProductID"}
```

In the example above, the Platform ID of S60 3rd Edition is used. The Product IDs of S60 3rd Edition devices are used in a similar way.

The Platform and Product IDs of S60 3rd Edition devices and newer are listed in Table 1 in Appendix A.



**Note:** From S60 3rd Edition onwards, the ID must be defined in square brackets because the platform dependency is considered a hardware dependency. If parentheses (dependency to a software component) are used, installation will fail with an "Application not compatible" error.

#### 3.2 Syntax in S60 1st and 2nd Editions

In S60 1st and 2nd Editions, the identifier sequence line is also added right after the line of the installation package header, but with the ID embraced by parentheses:

```
#{"MyApplication"}, (0x10000001), 1, 0, 0
; Platform ID for S60 2nd Edition
(0x101F7960), 0, 0, 0, {"Series60ProductID"}
```

In the example above, the Platform ID of S60 2nd Edition is used. The Product IDs of S60 1st and 2nd Edition devices are used in a similar way.

The Platform and Product IDs of S60 1st and 2nd Edition devices are listed in Table 2 in Appendix A.

## 4. Considerations for using Platform IDs and Product IDs

### 4.1 Applications supporting multiple platform releases

S60 C++ APIs are commonly targeted at multiple platform versions. Thus, it is worthwhile to design the application for multiple platform releases. Choosing the correct Platform ID helps in declaring the compatibility of each application build.

The general rule is that you should use the Platform ID of the first compatible S60 platform release to provide smooth installation on that and all subsequent platform releases (provided that there is no binary break between platform releases).

#### 4.1.1 S60 3rd Edition and later

Because of a full binary break in S60 3rd Edition, applications built for earlier platform versions are not compatible with it. In addition, the changed format of the installation package prevents the installation of incompatible applications. Thus, applications targeted to run on S60 3rd Edition or newer devices should always have a Platform ID of S60 3rd Edition (or newer if required).

The identifier for applications that are designed for S60 3rd Edition is `0x101F7961`. These applications can be installed on S60 3rd Edition and newer devices.

#### 4.1.2 S60 1st and 2nd Editions

Applications targeted to run on all S60 1st and 2nd Edition devices should use the Platform ID of S60 1st Edition (v0.9) (`0x101F6F88`).

Applications designed to run only on S60 2nd Edition devices should use the Platform ID of S60 2nd Edition (`0x101F7960`). It should then be possible to install such applications smoothly on all S60 2nd Edition devices (including all Feature Packs).

Similarly, if the application only runs on S60 2nd Edition, Feature Pack 2 and Feature Pack 3, then the Platform ID of S60 2nd Edition, Feature Pack 2 (`0x10200BAB`) should be used.

### 4.2 Device-specific applications

#### 4.2.1 Devices based on S60 3rd Edition and later

From S60 3rd Edition onwards, it is possible to define multiple Product IDs in the same .pkg file, to indicate support for multiple devices:

```
#{"DeviceSpecificApp"}, (0x20000001), 1, 0, 0
[0x10275218], 0, 0, 0, {"Nokia N71 ID"}
[0x200005F9], 0, 0, 0, {"Nokia N80 ID"}
```

An application with the sequence lines listed above installs to the two devices without warnings. The application will also install to any other compatible S60

3rd Edition (or newer) device but the installer will provide the user with a warning message: "Application not compatible with phone. Continue anyway?"



**Note:** Product IDs and S60 3rd Edition Platform ID should not be used together in a .pkg file if the application is meant to support only a subset of S60 3rd Edition devices. If the Platform ID is defined, the application will install to any device supporting that platform, without displaying a compatibility warning.

#### 4.2.2 Devices based on S60 1st or 2nd Edition

In S60 1st and 2nd Editions, the Platform ID is always required but you may combine it with one — and only one — Product ID:

```
#{"DeviceSpecificApp"}, (0x10000001), 1, 0, 0
(0x10200BAB), 0, 0, 0, {"Series60ProductID"}
(0x101F7964), 0, 0, 0, {"Nokia6630ProductID"}
```



**Note:** If multiple Product IDs are defined, the installation will fail. If the Platform ID is missing, a compatibility warning is displayed during installation.

## 5. Detecting the device at run time with the Machine UID

In addition to the identification codes, Machine UIDs are available for S60 C++ applications. An application can at **run time** find out the Machine UID to identify the device, and then optimize the available features to best suit the capabilities of the device running the application.

The Machine UID can be retrieved by calling the `HAL::Get` function in the source code:

```
#include <hal.h> // link against hal.lib

TInt uid;
HAL::Get(HAL::EMachineUid, uid);
```

Note that in newer S60 devices (from S60 2nd Edition, Feature Pack 3 onwards) the Machine UID is usually the same as the Product ID. All supported Machine UIDs are listed in Table 1 and Table 2.

## 6. Detecting the platform version during installation

As described in earlier chapters, all S60 application SIS packages contain a mandatory dependency to the Series60ProductID component. This dependency defines the minimum S60 platform version required in order to run the application, and prevents installation on older platforms. The Series60ProductID can be checked only during installation, and only for purposes of aborting the installation or warning about possible incompatibility. However, in some cases it would be useful to have a single SIS package that would select the installed files based on the S60 platform version. The following solution describes how to do this in a .pkg file. Similarly, a code snippet for checking the platform version at run time is provided.

Even though there is no dedicated attribute for the S60 platform version that could be used with IF condition blocks, it is still possible to resolve the version, based on the existence of specific files on the ROM drive.

Each Series60ProductID component is stored in a separate .sis file in the z:\system\install\ directory. These files are named as follows:

z:\system\install\Series60v3.1.sis	S60 3rd Edition, Feature Pack 1
z:\system\install\Series60v3.0.sis	S60 3rd Edition
z:\system\install\Series60v2.8.sis	S60 2nd Edition, Feature Pack 3
z:\system\install\Series60v2.6.sis	S60 2nd Edition, Feature Pack 2
z:\system\install\Series60v2.1.sis	S60 2nd Edition, Feature Pack 1
z:\system\install\Series60v2.0.sis	S60 2nd Edition
z:\system\install\Series60v1.2.sis	S60 1st Edition (1.2)
z:\system\install\Series60v1.1.sis	(n/a)
z:\system\install\Series60v1.0.sis	(n/a)
z:\system\install\Series60v0.9.sis	S60 v0.9

In each device, there is a corresponding file for the platform version it represents, and also files for all previous versions that are supported. For example, the Nokia N70 device (2nd Edition, FP3) contains all of the above files before Series60v3.0.sis, while the Nokia E70 device (3rd Edition) only has Series60v3.0.sis.

These filenames can be utilized in pkg files:

```
#{"MyApplication"}, (0x10000001), 1, 0, 0
; Lowest supported S60 platform version is 2nd Edition
  (0x101F7960), 0, 0, 0, {" Series60ProductID"}
; install common files for all supported platforms
```

```
; ...  
  
IF EXISTS("z:\system\install\Series60v2.8.sis")  
; install 2nd Ed, FP3-specific files  
  
ELSEIF EXISTS("z:\system\install\Series60v2.6.sis")  
; install 2nd Ed, FP2-specific files  
  
ELSEIF EXISTS("z:\system\install\Series60v2.1.sis")  
; install 2nd Ed, FP1-specific files  
  
ELSE  
; 2nd Edition-specific files  
  
ENDIF
```

## 7. Detecting the platform version at run time

### 7.1 Version Info API

From S60 3rd Edition, Feature Pack 2 onwards the platform version can be conveniently detected at run time using the Version Info API (`versioninfo.h`).

Class `TPlatformVersion` (inherited from `TVersionBase`) stores platform version information. The class is used as a parameter in `GetVersion()` methods. The edition and feature pack information is stored in two data fields:

```
TUint16 iMajorVersion
```

Contains information about the S60 Edition. For example, "3" in case of S60 3rd Edition, Feature Pack 2.

```
TUint16 iMinorVersion
```

Contains information about the Feature Pack. For example, "2" in case of S60 3rd Edition, Feature Pack 2.

```
#include <versioninfo.h>    // link against platformver.lib

// iFs is an open file server session (RFs)
VersionInfo::TPlatformVersion platformVersion;
TInt ret = VersionInfo::GetVersion( platformVersion, iFs );
if( ret == KErrNone )
{
    // platformVersion now contains version information
}
```

### 7.2 Detecting the version from .sis files

In earlier S60 editions, there was no dedicated S60 API for checking the platform version at run time (as there is for checking the device model, described in Chapter 5, "Detecting the device at run time with the Machine UID"). However, it is possible to use the following function to recover the S60 platform version. On return, `aMajor` and `aMinor` contain the highest version number found from the file names listed in Chapter 6. Note that while `aMajor` refers to the S60 Edition, `aMinor` cannot always be directly interpreted as a Feature Pack number (for example, 2.6 = 2nd Edition, FP2).

```
#include <f32file.h>    // link against efsrv.lib

_LIT(KS60ProductIDFile, "Series60v*.sis");
_LIT(KROMInstallDir, "z:\\system\\install\\");

void GetS60PlatformVersionL( RFs& aFs, TUint& aMajor, TUint& aMinor )
```

```
{
TFindFile ff( aFs );
CDir* result;
User::LeaveIfError( ff.FindWildByDir( KS60ProductIDFile,
    KROMInstallDir, result ) );
CleanupStack::PushL( result );
User::LeaveIfError( result->Sort( ESortByName|EDescending ) );
aMajor = (*result)[0].iName[9] - '0';
aMinor = (*result)[0].iName[11] - '0';
CleanupStack::PopAndDestroy(); // result
}
```

## Appendix A. Supported IDs in S60 devices

Platform	Device	Platform ID	Product ID	Machine UID
<b>S60 3rd Edition, FP1:</b>		<b>0x102032BE</b>		
	Nokia 5700 XpressMusic		0x20002D7C	
	Nokia 6110 Navigator		0x20002D7B	
	Nokia 6120 Classic / 6121 Classic		0x20002D7E	
	Nokia 6290		0x20000606	
	Nokia E90		0x20002496	
	Nokia E51		0x20002498	
	Nokia N76		0x2000060A	
	Nokia N81		0x20002D83	
	Nokia N95		0x2000060B	
	Nokia N95 8GB		0x20002D84	
<b>S60 3rd Edition:</b>		<b>0x101F7961</b>		
	Nokia 3250		0x200005F8	
	Nokia 5500 Sport		0x20000602	
	Nokia E50		0x20002495	
	Nokia E60		0x20001856	
	Nokia E61		0x20001858	
	Nokia E61i		0x20002D7F	
	Nokia E62		0x20001859	
	Nokia E65		0x20000604	
	Nokia E70		0x20001857	
	Nokia N71		0x200005FF	
	Nokia N73		0x200005FB	
	Nokia N75		0x200005FE	
	Nokia N77		0x20000601	
	Nokia N80		0x200005F9	
	Nokia N91		0x200005FC	

Platform	Device	Platform ID	Product ID	Machine UID
	Nokia N92			0x200005FA
	Nokia N93			0x20000600
	Nokia N93i			0x20000605

**Table 1: Supported IDs of S60 3rd Edition devices**

Platform	Device	Platform ID	Product ID	Machine UID
<b>S60 2nd Edition, FP3:</b>		<b>0x102032BD</b>		
	Nokia N70		0x10200F9A	
	Nokia N72		0x10200F9A	
	Nokia N90		0x10200F98	
<b>S60 2nd Edition, FP2:</b>		<b>0x10200BAB</b>		
	Nokia 6630		0x101F7964	0x101FBB55
	Nokia 6680		0x102078D1	0x10200F99
	Nokia 6681		0x102078D0	0x10200F9C
	Nokia 6682		0x102078CF	0x10200F9B
<b>S60 2nd Edition, FP1:</b>		<b>0x101F9115</b>		
	Nokia 3230		0x10200F97	
	Nokia 6260		0x101FB3F4	
	Nokia 6620		0x1020216B	0x101F3EE3
	Nokia 6670		0x101FD5DC	0x101FB3F3
	Nokia 7610		0x101FD5DB	0x101FB3F3
<b>S60 2nd Edition:</b>		<b>0x101F7960</b>		
	Nokia 6600		0x101F7963	0x101FB3DD
<b>S60 1st Edition, FP1 (1.2):</b>		<b>0x101F8202</b>		
	Nokia 3650		0x101F7962	0x101F466A
	Nokia 3660		N/A	0x101F466A
	Sendo-X		N/A	0x101FA031
	Siemens SX-1		0x101F9071	
<b>S60 1st Edition (0.9):</b>		<b>0x101F6F88</b>		
	Nokia 7650		0x101F6F87	0x101F4FC3
<b>Other:</b>				
	Nokia N-Gage™		0x101F8A64	0x101F8C19
	Nokia N-Gage™ QD		N/A	0x101FB2B1

**Table 2: Supported IDs of S60 1st and 2nd Edition devices and Nokia N-Gage™ game decks**

---

## Evaluate this resource

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