

Symbian Signed Q&As

Please note that changes introduced to Symbian Signed in March 2010 may have affected the information in this article. This article is in the process of being updated to reflect those changes. This message will be removed when those changes are complete.

What are Capabilities?

A Capability is a token that allows an application to access a protected resource. The Symbian OS Platform Security model defines 20 Capabilities. On Symbian OS, signing is the mechanism used to grant Capabilities to applications; via the signing process, developers request the Capabilities their application needs and, if granted, those Capabilities are then encoded into the digital certificate used to sign the application.

Symbian OS Capabilities are classified as either User or System Capabilities:

- User Capabilities are designed to be meaningful to mobile phone users and, for example, allow applications to send messages or access user data.
- System Capabilities protect system services, device settings, and hardware features.
- Restricted Capabilities are not available via all signing options.
- The most sensitive System Capabilities are referred to as Device Manufacturer Capabilities.

To access the Device Manufacturer Capabilities, you must complete a Capability Request Form and follow the instructions on the Device Manufacturer Capabilities Wizard.

More information on the process for Nokia is available here: [Capabilities](#).

Which APIs are Capability-protected?

On Symbian OS, Capabilities are used to protect APIs that allow sensitive operations. Examples include operations that may access end users' private data; that may create events that are billable; that may interfere with other applications; that may access the mobile phone network; or that may access handset functions that can affect the normal behavior of the phone. Do all Symbian OS applications have to be signed?

From Symbian OS v9, all applications that need to access protected APIs must have the appropriate Capabilities to do so. Approximately 40% of Symbian OS APIs are Capability-protected.

- Applications that use no protected APIs do not need Capabilities, and therefore may not need to be signed, depending on the security policy implemented on the handsets that the application targets. Note that the user will be warned that the software is not trusted and must agree to install the application.
- Applications that use only APIs protected by User Capabilities may not need to be signed, depending on the security policy implemented on the handsets that the application targets. The user will be warned that the software is not trusted and must agree to install the application. The user will also be asked at install time to grant the Capabilities the application requires.
- Applications that use any APIs protected by System Capabilities must be signed.

It is still possible to install unsigned Symbian OS applications on some phones (vendors decide on the security policy for each handset). However, there is no guarantee that Device Manufacturers will continue to allow unsigned applications to install on their devices. For commercial distribution therefore, and for any widescale distribution of freeware or shareware, signing should be considered a requirement. There is no requirement to sign applications targeted at versions of Symbian OS earlier than v9.

Do all phones based on Symbian OS support signed applications?

All phones based on Symbian OS releases from v9 fully support application signing. Phones based on earlier releases have either limited (v8) or no support for signing (v7 and earlier releases).

How much does it cost to get an application signed?

The costs of the different signing options vary.

A Publisher ID currently costs \$200 annually.

Each submission through Express Signed costs €10.

Each submission through Certified Signed will cost the fee as set by the [Test House](#). At the time of writing, the test house charge €150 per submission.

Open Signed Online for testing your application is free.

Will I have to pay again for each new release of my application?

Yes, each time you sign software you will incur the costs which apply to the signing option you have chosen.

What happens if my application fails testing?

The Test House will provide a test report identifying the reasons for failure. You can track the progress of your application via your Symbian Signed account. For Certified Signed you will need to resolve the issues, and resubmit your application for testing. You will have to pay to have your application re-tested, therefore you are advised to test your application thoroughly against the Test Criteria before your initial submission to reduce the likelihood of failure. Re-submitting an application for testing may cost less than the initial submission, but this is test house dependent.

For Express Signed the submitter may be required to submit future applications via Certified Signed only, for some period or for some number of signings. The failed application may also be revoked.

Does Symbian Signed apply to Java MIDlets?

No, Symbian Signed only applies to software which is distributed in SIS file format; developers of Java MIDlets should sign their applications using Sun's Java Verified scheme.

Signing does apply to software written in other languages, for example, Flash or Python, and is distributed in SIS file format.

I have UIDs allocated from uid@symbiandevnet.com. Can I still use them?

A requirement for getting a Symbian OS v9 application signed is that the UID comes from the new system and is in the protected range. Even if you have previously obtained a UID from Symbian it will be necessary to reapply at <https://www.symbiansigned.com/app/page> regardless of whether or not you are intending to sign your application.

You can also continue to use your existing allocations for unsigned application usage on Symbian OS v9. To do this, simply replace the first hex digit (a 1) with F, and leave the remaining digits unaltered. This maps your UID into the Legacy UID compatibility range where it will not conflict with any other allocations. For example, you have a UID allocation 0x100F55BE which you can transpose to 0xF00F55BE for use in an unsigned Symbian OS v9 application.



© 2010 Symbian Foundation Limited. This document is licensed under the [Creative Commons Attribution-Share Alike 2.0 license](http://creativecommons.org/licenses/by-sa/2.0/legalcode). See <http://creativecommons.org/licenses/by-sa/2.0/legalcode> for the full terms of the license.

Note that this content was originally hosted on the Symbian Foundation developer wiki.

