How to guide for creating/signing sis files

This article explains about creating and signing `.sis` files. It applies to Symbian OS 9 (or newer) and S60 3rd Edition (or newer). For earlier Symbian OS or S60, signing is not mandatory.

This article needs to be updated: If you found this article useful, please fix the problems below then delete the {{ArticleNeedsUpdate}} template from the article to remove this warning.

Reasons: hamishwillee
(29 Nov 2011)
The article content is useful and accurate. However there are heaps of articles on the wiki with the same topic, some that also cover using the IDE, carbide, Qt Creator, using devcerts, using local certificates, using createsis. I think this needs to be merged with the others.

Creating `.sis` files

A `.sis` file is created from `.pkg` file using makesis tool, a standalone command line tool. For example:

```
makesis myapp.pkg
```

The command above creates `myapp.sis` from `myapp.pkg`.

In case of your PKG-file was generated by Carbide.c++, you will get error, something like this:

```
 Error : Cannot find file $(EPOCROOT)Epoc32\release\$(PLATFORM)\$(TARGET)...
```

To avoid this error, you should replace macroses `$(EPOCROOT)`, `$(PLATFORM)`, `$(TARGET)` with the real pathes to the files.

Signing `.sis` files

Signing a `.sis` file can be done using signsis tool. For example:

```
signsis myapp.sis myapp.sisx mycert.cer mykey.key
```

The example above signs `myapp.sis` into `myapp.sisx` using the certificate of `mycert.cer` and the private key of `mykey.key`.

Note that some developers give the extension of `.sisx` for signed file. However, it does not mean that `.sis` is always unsigned file. There are many `.sis` file that is signed.

There are several certificates that can be used for signing, i.e.:

- Self-signed certificate
- Developer certificate
- Symbian Signed certificate

Self-signed certificates

An application can be signed using self-signed certificate if it does not require any capabilities that cannot be granted by the users. Technically, it means the application does not require any capability or require one (or more) of the following capabilities:

- LocalServices
- Location (starting with S60 3rd Edition FP2)
- NetworkServices
- UserEnvironment
- ReadUserData
- WriteUSerData

Self-signed certificate is generated by the developer and as such it cannot be trusted by the device. During installation, the users will be prompted a warning dialog saying that the application is not trusted.

This type of certificate can be used for signing commercial releases of the applications.

Developer certificate

An application has to be signed with developer certificate if it requires more than the basic capabilities above. Developer certificate is used in the testing and development phase to be able to test the application on a real device. The reason is because the certificate is bound to the IMEI. An application signed

(C) Copyright Nokia 2013. All rights reserved.

with this certificate can only be installed on a device whose IMEI is supported by the certificate.

One way to apply for Developer Certificate is via Symbian Signed. Some phone manufacturers may have their own developer certificates as well.

This type of certificate cannot be used for signing commercial releases of the applications.

Symbian Signed certification

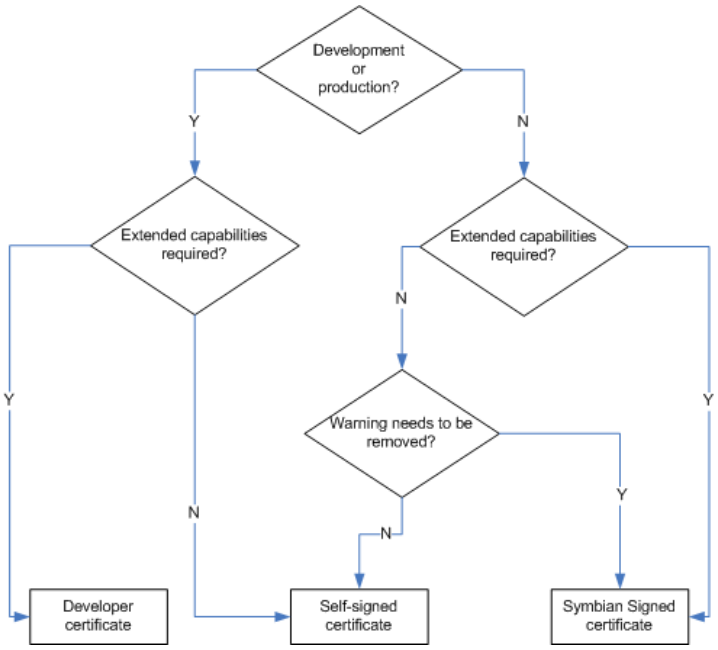
Symbian Signed certification is mandatory for the application that requires extended of set capabilities, i.e. capabilities that cannot be granted by the users during installation. It is intended for public distribution, not for development. Unlike developer certificate, an application signed with this certificate can be installed on any compatible device. There is not IMEI limitation.

There are a number of requirements that have to be fulfilled to get certification. Symbian Signed performs a set of test case to the application before it can be signed with Symbian Signed certificate. The Symbian Signed certification process costs a few hundred Euros. Please check the latest requirements and cost of getting certification from Symbian Signed web site.

Note: Self-signed application does not need to go to Symbian Signed certificate. It can go through Symbian Signed if developers want to eliminate the warning dialog during installation.

Which certificate to use?

The picture below shows how to choose which certificate to use.



- The first question that needs to be answered is whether the signing is for development or production.
- If it is for development, it can be self-signed certificate or developer certificate depending on the required capabilities. Note that, developer certificate can be used in both cases.
- If it is for production, an application normally goes to Symbian Signed. The exception is when the developer does not mind distributing it as untrusted application. In which case, self-signed certificate can be used.

Batch file for creating and signing sis file

When using command line a good is to create batch file Once you have the certificate and the key file ready for your application use the following batch file to create and sign the sis file.

```
makesis mypkg.pkg myappunsigned.sis
signsis -v myappunsigned.SIS myapp.sis appcertificate.cer appkey.key mypass
```

- Just copy the above two lines in notepad and name the file as sign.bat
- Copy this file in your SIS folder and whenever you need to sign just type sign.bat from the command prompt.

Internal links

- Archived:How to sign a .Sis file with Self-Sign Certificate
- Developer Certificate (Symbian Signed)
- Freeware signing
- How do I use the same pkg file for debug and release builds?

(C) Copyright Nokia 2013. All rights reserved.

External links

- Symbian Signed

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.