

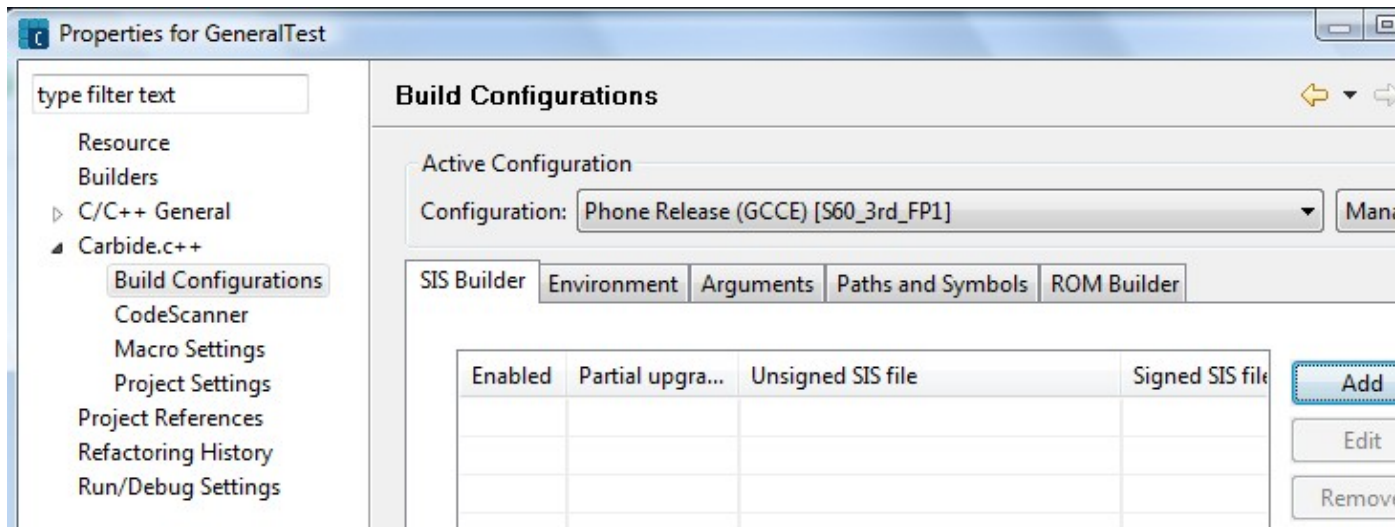


Note!

Carbide.c++ IDE offers inbuilt feature to create and sign the .sis file with Self-signed certificate or Developer certificate. The images below:

- Step1 : Open your project in Carbide.c++

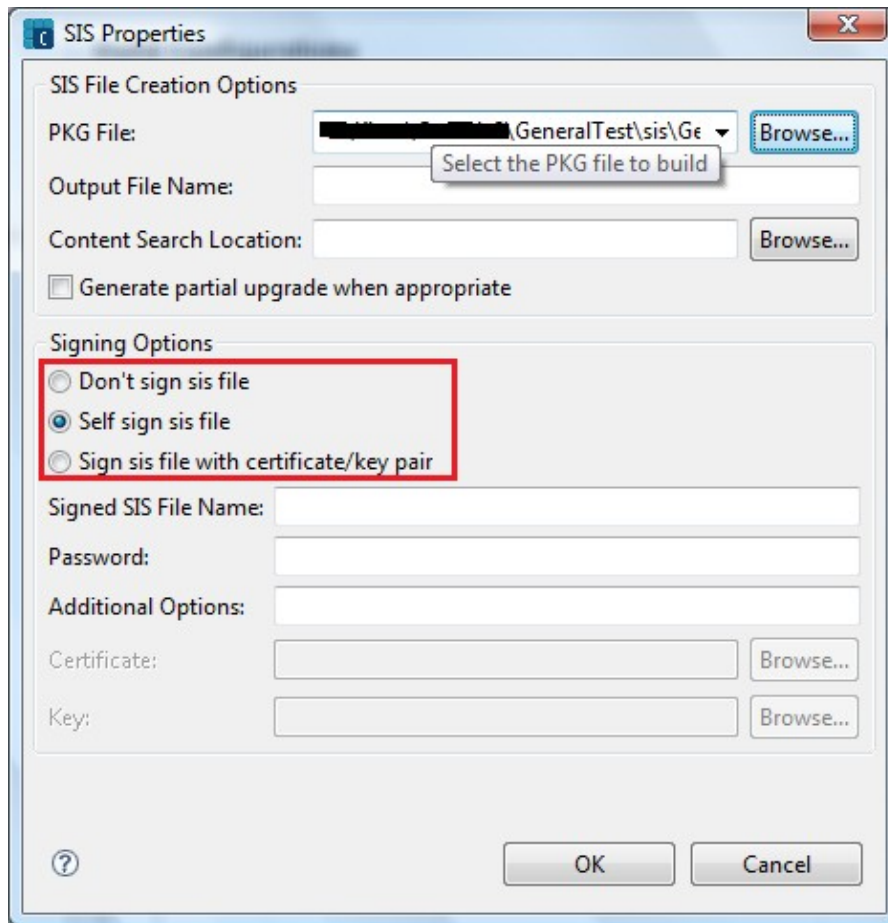
Project >> Properties >> Carbide.c++ >> Build Configurations >> SIS Builder



- Step2 :

Project >> Properties >> Carbide.c++ >> Build Configurations >> SIS Builder >> Add

How_to_sign_a_.Sis_file_with_Self-Sign_Certificate



This article is specific to S60 3rd Edition development or newer. From S60 3rd Edition, which is based on Symbian OS 9, a user can install only signed applications.

If an application requires basic capabilities only, it can be signed with self-signed certificate. Basic capabilities include LocalServices, NetworkServices, ReadUserData, UserEnvironment and WriteUserData. Starting with S60 3rd Edition Feature Pack 2 release (aka S60 3.2) the Location capability can also be granted to self-signed applications.

An application that does not require any capabilities can also be signed with self-signed certificate. The advantage of self-signing is that it's fast, since it's done locally on developer's machine and that a self-signed installable package can be installed on more than one device, i.e. no restriction applies (based on IMEI number, for example).

The drawback of self-signed certificate is that a user will be prompted a warning dialog. It says that the application is not trusted and the user can decide whether or not to continue the installation. The warning dialog does not appear when the application is signed via Symbian Signed.

Please note that self-signing works even if the application demands stronger capabilities than what is in the basic set, however, it won't install on the device.

How_to_sign_a_.Sis_file_with_Self-Sign_Certificate

Aside from the capability restrictions please note that you cannot self-sign applications having the UID3/SID in the protected range. In practice, during R&D stage you can use a random UID from the 0xE***** range but for the application release you must still use an UID allocated to you, for free, by Symbian. This UID will be in the 0xA***** range and can be obtained through the [Unprotected UID Allocation Request](#) form.

Following is the example which illustrates the Self-Signing procedure.

Step-1 Creating Certificate and Keys

From your Desktop Screen: **Start-> Run -> Cmd** then go in your **Sis** file path using **cd** DOS command.

```
C:\HelloWorld\sis> makekeys -cert -password World123 -len 1024 -dname "CN=World User OU=Development"
```

Note: It is important the the above code be written on one line. This applies to the code below too.

This will create **WorldKey.key** and **WorldCert.cer** files which you will need to sign your **HelloWorld.Sis** file in the following statement.

Important: Versions of MakeKeys prior to Symbian 9.2 always generate certificates with a validity period of 1 year. This may be too short for released code, as it restricts the lifetime of the signed SIS files. You can [create a certificate using OpenSSL](#) to choose a different validity period, or use the update to MakeKeys provided by Symbian [here](#) to specify an alternative validity period.

Step-2 Signing Application

```
C:\HelloWorld\sis>signsis HelloWorld.sis HelloWorld.sis WorldCert.cer WorldKey.key World123
```

Note: Here you may find that **World123** is the password.

Finally you will get your Self-Signed certified **HelloWorld.Sis** file. Install it on the device. Have fun !!!

Related articles

[S60 SW installer troubleshooting](#)