

# DevCertRequest User Guide

1.	<b>INTRODUCTION</b> .....	2
2.	<b>PROCESS OVERVIEW</b> .....	2
3.	<b>INFORMATION DEFAULTING</b> .....	2
4.	<b>DEVCERTREQUEST STEPS</b> .....	3
	STEP <b>1</b> : SELECT CERTIFICATE REQUEST (.CSR) FILENAME .....	3
	STEP <b>2</b> : INPUT PRIVATE KEY AND PASSWORD .....	4
	STEP <b>3</b> : PERSONAL DETAILS .....	4
	STEP <b>4</b> : CAPABILITY AND IMEI SELECTION .....	5
	4.1. IMEI EDITOR .....	6
	4.1.1. <i>Importing IMEIs from file</i> .....	6
	STEP <b>5</b> : GENERATE CERTIFICATE REQUEST (.CSR) FILE .....	7
5.	<b>WARNING AND ERROR DIALOGS</b> .....	8
	5.1. PUBLISHER ID LOAD FAILURE .....	9
6.	<b>HANDLING OF LARGE NUMBER OF IMEIS</b> .....	9
7.	<b>SIGNING A SIS FILE WITH YOUR DEVELOPER CERTIFICATE</b> .....	9
8.	<b>GENERATEIMEIS.BAT UTILITY TOOL</b> .....	10
9.	<b>INSTALLATION</b> .....	11
10.	<b>FURTHER INFORMATION</b> .....	12
11.	<b>DOCUMENT HISTORY</b> .....	12

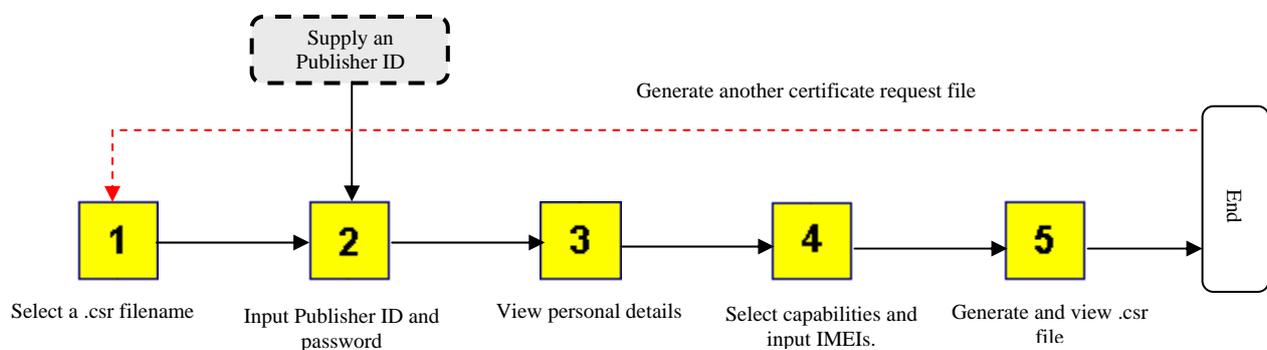
## 1. Introduction

DevCertRequest is a Microsoft Windows based application wizard that generates Symbian OS compliant Content Signing Request (.csr) files used for subsequently requesting Symbian Developer Certificates from the [www.symbiansigned.com](http://www.symbiansigned.com) portal.

This User Guide explains the steps involved during the process of generating a certificate request file (.csr) when using the wizard. Please refer to the <http://developer.symbian.com/main/signed/> for further information. Note that you must be registered at [SymbianSigned](http://www.symbiansigned.com) to request Symbian Developer Certificates.

## 2. Process Overview

Generating a certificate request file involves five necessary and distinct steps. The wizard guides the user through each step in sequence, where subsequent steps are not executed unless all user input pertaining to the current step have been supplied satisfactorily.



Each cycle generates a single .csr containing the requested capabilities and IMEIs. Once created, the .csr file cannot be updated, and a new one must be regenerated if any changes are required. You must purchase a TCT publisher ID, more information can be found here about the [TCT Publisher ID](#).

Use the  button to go back to previously executed steps and change details at anytime. Further .csr files can be generated by repeating the process without exiting the wizard.

## 3. Information defaulting

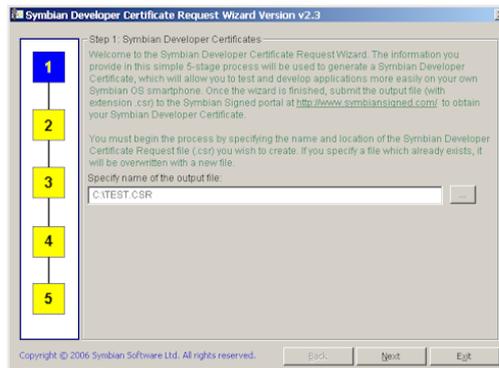
DevCertRequest saves the last entered personal details (SSLStoreDefault.txt), IMEIs and selected capabilities in a file. Relevant display fields are preloaded with information from this file for Step 3 and 4 on step entry. Additionally, the wizard also creates an information .cnf file (that contains the .csr contents in a readable form) for every successfully generated .csr file, and the AppLog.txt file - which is (along with the SSLStoreDefault.txt) saved in the launch directory.

Use the  button (where available) to clear any previously entered information.

## 4. DevCertRequest Steps

### Step **1** : Select Certificate Request (.csr) filename

Choose the filename and the directory for the .csr file to be generated. Use the file browse  button to select an existing file, or create a new one by pre-pending the name to the .csr extension in the file browse dialog.



Press the  button to continue.

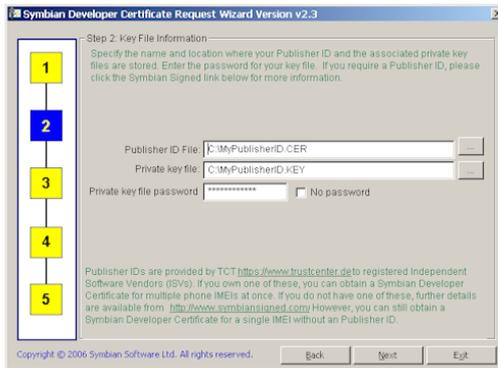
#### Notes:

1. If selected, an existing .csr file will be overwritten in Step 5.
2. You can optionally type just the certificate request filename (without the extension, if desired) in the above input box. In that case, if the path is not supplied, DevCertRequest will create the (.key) file in the directory where it was launched from. However, it is recommended to always select a single directory to house all .csr (and .key) files for ease of access. If a new filename is entered at a later stage, the same path is used as previously entered path.
3. Internally, DevCertRequest uses short form of directory and file names; hence, the filename displayed will be shortened and look different to the one originally entered. This scheme applies to all file entry fields in DevCertRequest.

## Step 2 : Input private key and password

This step affects and imposes constraint on what can be entered in the subsequent steps. You have the following options...

### 1. Publisher ID:



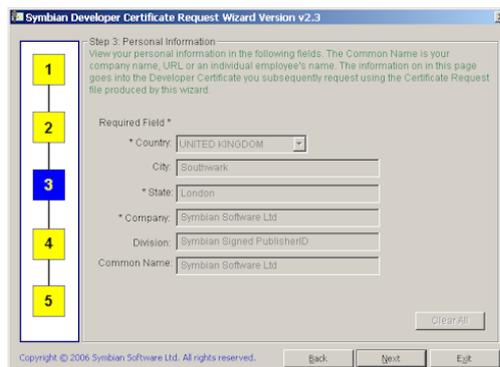
- Use the file browse  button to select the Publisher ID filename and the associated private key file names in the appropriate boxes. The files must exist before moving to Step 3.
- Uncheck  No password and enter the Publisher ID private key password, if key is protected. Make sure you select the appropriate key belonging to the Publisher ID.

Supplying an Publisher ID enables:

- Automatic preloading and displaying of Publisher ID personal information for Step 3.
- Entering of more than 1 non-repeated IMEIs.
- Selection of more (including Phone Manufacturer approved) capabilities.

For more information please see the FAQ on <http://developer.symbian.com/wiki/display/sign/Symbian+Signed>

## Step 3 : Personal Details



Personal details are automatically preloaded from the Publisher ID and displayed.

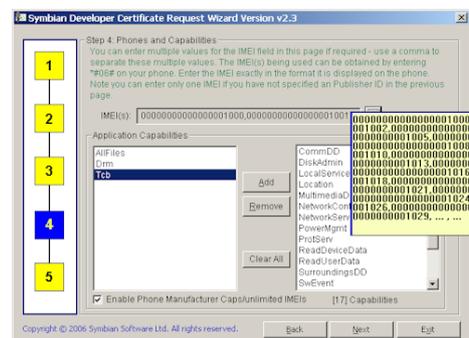
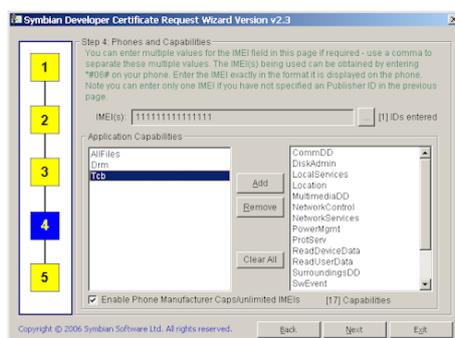
Press the  button to continue.

**Notes:**

1. Previously entered personal information is overwritten and saved as default with the current information (which could be from the Publisher ID).
2. The Common Name (CN) is defaulted to the company name if not supplied.
3. The personal details are viewable during SIS file installation. Ensure the details are correctly entered before generating the .csr file. Note that personal details are aggregated and referred to as a Distinguished Name (**DN**) within a certificate.

**Step 4 : Capability and IMEI selection**

A context sensitive IMEI and capability input step. The number of allowed IMEIs and capabilities is dependent upon the availability of the Publisher ID. For further information please see <http://developer.symbian.com/main/signed/>



- Check the  **Enable Phone Manufacturer Caps/unlimited IMEIs** box if Phone Manufacturer approved capabilities and more than 1000 IMEIs are to be entered. You will be required to authorize the generated certificate request file prior to requesting a Developer Certificate by the appropriate authority. This is necessary only if Phone Manufacturer approved capabilities are actually selected and/or more than 1000 IMEIs are entered.
- Select and **Add** the required capabilities from the left-hand capability selection list to the right-hand capability list and **Remove** previously selected capabilities, if necessary. You can add multiple capabilities in one 'Add' operation; however, only one capability at a time can be removed from the right-hand capability list.
- Click the **...** button to Add/Edit/Remove IMEIs. See IMEI Editor Section for more details. You can get IMEI summary list if you 'hover' the mouse pointer above the read-only IMEI text box, as shown in previous screenshot.

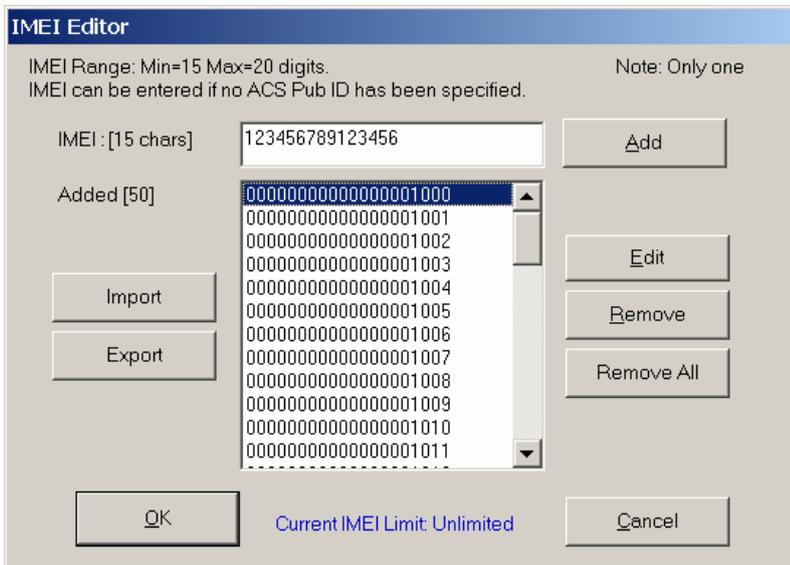
Press the **Next** button to continue.

**Notes:**

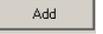
1. Non-Phone Manufacturer approved capabilities are pre-selected in the left-hand capability selection list when entering Step 4 or '(un)checking' the  **Enable Phone Manufacturer Caps/unlimited IMEIs** box. Phone Manufacturer approved capabilities are never pre-selected and require individual selection.
2. Right-click mouse button to select and/or unselect all capabilities on the left-hand capability selection list.
3. Previously entered excess capabilities are cleared if the  **Enable Phone Manufacturer Caps/unlimited IMEIs** box is unchecked. See section [Warning and Error Dialogs](#) for further details on IMEI/Capability mismatch.

## 4.1. IMEI Editor

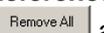
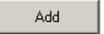
The editor provides a user-friendly interface for entering the desired IMEIs. Use the IMEI Editor to Add/Remove/Edit required IMEIs individually or from a file. The editor is context sensitive; hence, IMEI limitations apply. [FAQ-1136](#) briefly explains the IMEI format.



It is imperative to enter IMEI as seen after typing `*#06#` on the phone. Note that some phones omit the ('-') in the IMEI. Entering an IMEI incorrectly will result in the SIS file being rejected during installation. The current IMEI limit in-force can be seen at the bottom of the editor screen in [blue](#).

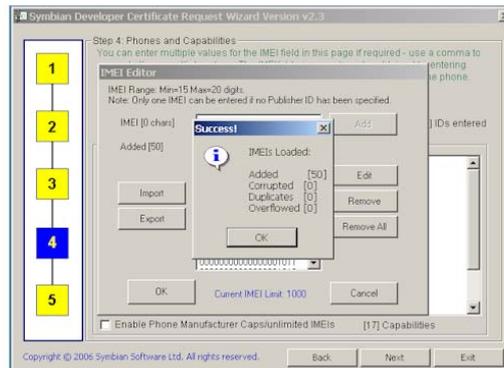
- Type the non-alpha (that can only include hyphens '-') IMEI in the IMEI input edit box. The  button will be enabled automatically when the IMEI being entered is within the IMEI length range. Press the  button to add the IMEI. The IMEI format is verified and the user is notified if the IMEI is non-conformant or a duplicate.
- Use  and  buttons accordingly on a selected IMEI from the list.

### Notes:

1. You can press  to discard the last entered IMEIs. Previously saved IMEIs are preserved.
2. When inputting an IMEI,  and  buttons are automatically disabled (and re-enabled when selecting an IMEI from the list).
3. When the IMEI limit is reached, the IMEI input box,  and  buttons are automatically disabled. You can only ,  or  IMEIs when the limit has reached - unless the IMEI count returns below the limit.
4. If at the time of launching the editor, the IMEI count exceeds (see section **Error! Reference source not found.** of when that can happen) the current IMEI limit, all except ,  and  buttons are enabled. Excess IMEIs must be removed before adding or editing any IMEIs.
5. Pressing the 'Enter' key when the  button is enabled, and the IMEI input edit box is in focus (i.e. the cursor is within the box), will add the IMEI in the IMEI list. In all other cases, pressing the 'Enter' key will dismiss the IMEI Editor dialog and return you back to the Step 4 dialog.

### 4.1.1. Importing IMEIs from file

In addition to inputting single IMEIs, the IMEI Editor also allows importing IMEIs from an ASCII text file.



See sample file below. Use a simple editor e.g. notepad.exe to create and edit the IMEI file. Every IMEI should be listed in a new line. Comments using '#' and empty lines are ignored during parsing. The editor verifies each IMEI in-turn, and either adds or discards (duplicates, corrupt and/or overflowed) IMEIs.

```
# Valid IMEIs
12345678-123456-1
12345678-123456-2
12345678-123456-3
123456781234561
123456781234562

# Corrupt IMEIs
12345678
12345-78-12-456-1

# Duplicate IMEIs
12345678-123456-2
123456781234562
```

- Click on the  button and select the file to import.
- Wait for the import to complete.
- The editor creates an import summary log file if any duplicates, corrupted and/or overflowed IMEIs are encountered. Press the  button to view the file or  to delete the log file:

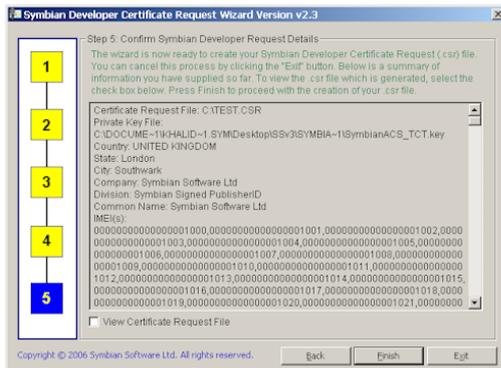
c:\temp\IMEI\_Log.txt

**Notes:**

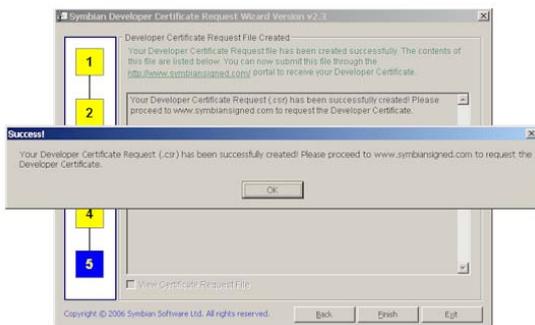
1. If the IMEI limit is reached after the import, point 4.1.3 will apply. Discarded excess IMEIs are marked as overflowed in the log file.
2. The  button is automatically disabled when the current IMEI limit has been reached.
3. IMEI ranges cannot be specified. Each IMEI must be entered individually in a new line.
4. Leading and trailing IMEI spaces are trimmed before verifying and adding the IMEI.

**Step 5 : Generate certificate request (.csr) file**

This is a confirmation and .csr generation step. The dialog displays a summary of the information entered. You can go back to the previous steps if you wish to amend any details.



You can view the .csr file contents once it has been generated. Press the **Finish** button to generate the certificate request file.



The dialog confirm successful certificate request file generation – unless an error was encountered, in which case the user is notified accordingly. Check the .csr (and the .key) files have been created in the desired directory. Ensure the private key remains private. Use the generated certificate request (.csr) file to request a Developer Certificate from the Symbian Signed portal.

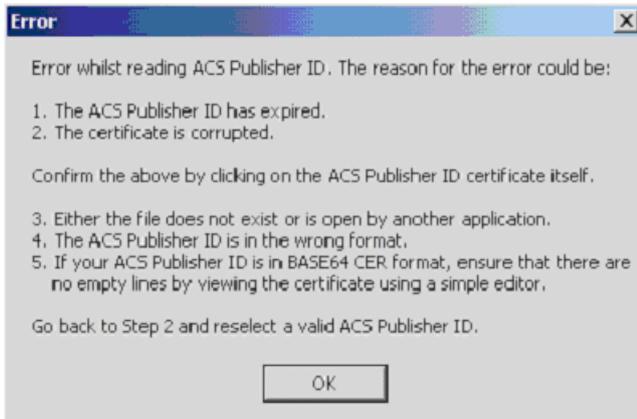
Press the **Start** button to generate another .csr file or close the wizard.

## 5. Warning and Error Dialogs

DevCertRequest issues warning/error dialogs in certain (rare) situations. The instructions in the dialogs require attention prior to proceeding to the next step. This section highlights these warning/error dialogs and what steps to take.

## 5.1. Publisher ID load failure

If an error is encountered when loading information from the Publisher ID, the screen in Step 3 will appear as follows:



All fields will be set with the 'Not Available' text and you will not be able to proceed to the next step. The reason for the failure could be:

- The Publisher ID has expired.
- The certificate is corrupted.

Confirm the above by clicking on the Publisher ID certificate itself.

- If your Publisher ID is in BASE64 CER format, ensure that there are no empty lines by viewing the certificate using a simple editor.
- Either the file does not exist or is open by another application.
- The Publisher ID is in the wrong format.

Go back to Step 2 and reselect a valid Publisher ID.

## 6. Handling of large number of IMEIs

DevCertRequest has been designed to handle inputting and handling of large number of IMEIs. As parsing and processing of such large number of IMEIs can take some time, DevCertRequest will change the mouse pointer to  during step transitioning or when importing IMEIs. The text 'Updating...' might also appear temporarily as shown above during entering or leaving some steps; indicating the processing of large number of IMEIs.

## 7. Signing a SIS file with your Developer Certificate

In Symbian OS v9, the process of signing and creating a SIS file has been decoupled. In Pre-v9 Symbian OS the application .pkg file included the certificate and the private key for signing the SIS file. The 'makesis.exe' command created the SIS file and subsequently signed it with one command. As of v9, the 'makesis.exe' command only creates a SIS file, whilst the new 'signsis.exe' tool does the signing of SIS files. The SIS file is created first followed by signing.

1. Make sure you have the correct Developer Certificate to use for signing. See the Symbian Signing <http://developer.symbian.com/wiki/display/sign/Symbian+Signed> for troubleshooting application installation and running issues.
2. Set your default device using [devices.exe](#).

3. Create the SIS file for the application using makesis.exe. E.g.
  - `makesis -d%EPOCROOT% myapp.pkg`
  - See [FAQ-1113](#) for using the '-d' option.
4. Copy the appropriate Developer Certificate and key files to where the SIS file was created.
5. Run the 'signsis.exe' as follows:
  - `signsis myapp.SIS myapp_signed.SIS myDevCert.cer myDevCert.key password`
  -

This will sign the SIS file with the Developer Certificate. Use the same password as when the key was created using DevCertRequest.exe.
6. Install the SIS file on the target device.
7. Test the application as required.

## 8. GenerateIMEIs.bat utility tool

DevCertRequest is accompanied with a simple tool to generate multiple IMEIs to a file or console. Symbian Signed use this tool internally to test DevCertRequest and the portal for large number of IMEIs prior to a new release. The script is written in Perl (although, as the file extension indicates the Perl commands are wrapped in a BATCH file) and installs as part of standard DevCertRequest installation (see below).

You can use this tool to generate sequential IMEIs for your use, and either output the IMEIs to a file or alternatively, the console. You can then import these IMEIs into DevCertRequest via the 'import' feature in the IMEI Editor 4.1. Beside other options (see `GenerateIMEIs.bat -h`) the tool allows you to specify the start value, in addition to the length of the IMEIs to generate. Output file name takes the form of `IMEIs_<imeicount>.txt`. The following page explains some common usage examples:

1. To generate 10 IMEIs starting from '0' and IMEI length of 20 digits run the command as below. Default behaviour is to output the IMEIs into a file which is created where the tool was executed from.
  - `GenerateIMEIs.bat 10`
2. Extension to the above but this time the start value is specified and outputting is to the console. The output from the console can be redirected to a file, if desired. By running the command again to generate a different set of IMEIs using '-c' console option, different set of IMEIs can be augmented into one file (using append redirection '>>').
  - `GenerateIMEIs.bat 20 10000000 -c`
  - `GenerateIMEIs.bat 20 10000000 -c -d >> myIMEIs.txt`
3. As the maximum number of digits that can appear in an IMEI is twenty, the unused most-significant digits are padded with '0'. If you wish to limit the number of digits or remove any padding, the '-l' option can be used to specify the IMEI length. Note that the tool will display an error if the IMEIs to be generated result in a length greater than the supplied length or the maximum twenty-digit limit – although, the tool will allow you to generate IMEIs smaller in length than the minimum fifteen-digit limit. By generating smaller length IMEIs you can manually amend the IMEIs in the output, if you require non-sequential IMEI range(s).
  - `GenerateIMEIs.bat 20 -d -l 10`
  - `GenerateIMEIs.bat 30 10000000 -d -l 16`

If you wish to supply your own output filename then the only way to do this is to redirect the output of the console (using '-c' option) to a file and directory of your choice. The '-o' does not accept a filename. The tool generates the output file name using the above mentioned filename format. This makes it simpler to delete ('-d') any previously generated output files.

The tool also provides the '-r' option to report essential statistics for a given IMEI input file. Use this option to verify an IMEI file that was generated previously before importing into DevCertRequest. In addition to outputting total IMEI count and other simple statistics, the '-r' option also displays the ranges of IMEIs found in an input file.

## 9. Installation

You can upgrade DevCertRequest if installing to a higher version – otherwise, you will be forced to uninstall DevCertRequest if reinstalling the same version.

The utility tool `GenerateIMEIs.bat` installs to `c:\Program Files\Common Files\Symbian\tools` by default. This is where all Symbian stub build tools are extracted when you install a Symbian OS based SDK e.g. UIQ, S60 etc. When you install a SDK, the environment variable `PATH` gets updated automatically to include the above path; hence, you can run this utility tool from any drive or directory. Alternatively, you can start the tool from the 'Start' bar by navigating to the DevCertRequest menu. A `cmd.exe` window is launched in the above directory from where the tool can be executed.

## 10. Further Information

1. For Symbian OS information <http://www.symbian.com>.
2. <https://www.symbiansigned.com> portal.
3. Symbian OS (and Symbian Signed) related [FAQ](#) knowledgebase.
4. For the complete process please see <http://developer.symbian.com/main/signed/>
5. A Symbian forum discussion.symbian.signed has been created and it can be found at the following NNTP Server: publicnews.symbiandevnet.com. For further information click this link [NewsGroups](#).
6. For quick reference in Symbian Signing, please see [http://developer.symbian.com/main/downloads/files/Symbian\\_Signed\\_Grid.pdf](http://developer.symbian.com/main/downloads/files/Symbian_Signed_Grid.pdf)

## 11. Document History

Date	Version	Status	Author	Description
15-01-2008	1.9	Issued	KM	1. Modified to support DevCertRequest 2.3
14-07-2007	1.8	Issued	KM.	2. Updated links and some wording 3. Changed 100 IMEI's to 1000 4. Replaced Images
23-10-2006	1.7	Issued	KM.	5. Updated links and some wording 6. Changed 20 IMEI's to 100 7. Added newsgroups.
03-05-2006	1.6	Issued	Nishaat R.	8. Minor changes to section 4.1.1 – to reflect the location change for IMEI error log file. 9. Added section Handling of large number of IMEIs.
06-04-2006	1.5	Issued	Nishaat R.	1. Added usage information on the utility script GenerateIMEIs.bat. 2. Other minor changes.
24-11-2005	1.4	Issued	Nishaat R.	Screenshot updates for Step 3.
15-11-2005	1.3	Issued	Nishaat R.	1. Minor changes to screenshots. 2. Added information about optional fields for personal information in DevCertRequest.
17-10-2005	1.2	Issued	Nishaat R.	Minor User Guide updates.
09-09-2005	1.1	Issued	Nishaat R.	Screenshot and other minor text updates.
11-08-2005	1.0	Issued	Nishaat R.	Final changes before first issue. Changes reflect more DevCertRequest testing comments from internal stakeholders. Fixed links and typos.
29-06-2005	0.3	Draft	Nishaat R.	Further updates after further comments and changes to DevCertRequest.
17-06-2005	0.2	Draft	Nishaat R.	Update after first internal review.
15-06-2005	0.1	Draft	Nishaat R.	Initial draft sent for review.