



MOTODEV

The Motorola developer network

UIQ/SYMBIAN DEVELOPMENT
V 1.0

TECHNICAL ARTICLE



Copyright © 2008, Motorola, Inc. All rights reserved. This documentation may be printed and copied solely for use in developing products for Motorola products. In addition, two (2) copies of this documentation may be made for archival and backup purposes. Except for the foregoing, no part of this documentation may be reproduced or transmitted in any form or by any means or used to make any derivative work (such as translation, transformation, or adaptation) without express written consent from Motorola, Inc.

Motorola reserves the right to make changes without notice to any products or services described herein. "Typical" parameters, which may be provided in Motorola Data sheets and/or specifications, can and do vary in different applications and actual performance may vary. Customer's technical experts will validate all "Typicals" for each customer application.

Motorola makes no warranty in regard to the products or services contained herein. Implied warranties, including without limitation, the implied warranties of merchantability and fitness for a particular purpose, are given only if specifically required by applicable law. Otherwise, they are specifically excluded.

No warranty is made as to coverage, availability, or grade of service provided by the products or services, whether through a service provider or otherwise. No warranty is made that the software will meet your requirements or will work in combination with any hardware or application software products provided by third parties, that the operation of the software products will be uninterrupted or error free, or that all defects in the software products will be corrected.

In no event shall Motorola be liable, whether in contract or tort (including negligence), for any damages resulting from use of a product or service described herein, or for any indirect, incidental, special or consequential damages of any kind, or loss of revenue or profits, loss of business, loss of information or data, or other financial loss arising out of or in connection with the ability or inability to use the Products, to the full extent these damages may be disclaimed by law.

Some states and other jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, or limitation on the length of an implied warranty, therefore the above limitations or exclusions may not apply to you.

This warranty gives you specific legal rights, and you may also have other rights, which vary from jurisdiction to jurisdiction.

Motorola products or services are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product or service could create a situation where personal injury or death may occur.

Should the buyer purchase or use Motorola products or services for any such unintended or unauthorized application, the buyer shall release, indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the designing or manufacturing of the product or service.

Motorola recommends that if you are not the author or creator of the graphics, video, or sound, you obtain sufficient license rights, including the rights under all patents, trademarks, trade names, copyrights, and other third party proprietary rights.

If this documentation is provided on compact disc, the other software and documentation on the compact disc are subject to the license agreement accompanying the compact disc.

UIQ/Symbian Development

Version 1.0

July 2008

For the latest version of this document, visit <http://developer.motorola.com>

Motorola, Inc.

<http://www.motorola.com>

Contents

An introduction to UIQ/Symbian development	4
Background	4
Commercial development steps.....	4
Non-commercial development steps.....	5
Developing UIQ applications	7
Register at UIQ Developer Community.....	7
Download UIQ SDK	7
Install SDK	8
Download MOTODEV Studio for UIQ.....	9
Install MOTODEV Studio for UIQ.....	9
Building a native application	10
Register at Symbian Signed	10
Request application UID	12
Modify HelloWorld.....	14
Build HelloWorld via command line	14
Using MOTODEV Studio for UIQ to build HelloWorld	15
Using the emulator.....	19
Launching HelloWorld using the command line.....	19
Launching HelloWorld using the IDE	20
Modifying and testing the HelloWorld application	21
Signing and packaging an application	22
Signing overview	22
Signing capabilities	22
Signing options.....	23
Open signed	23
Open signed online.....	23
Open signed offline.....	23
Express signed	24
Certified signed.....	24
Packaging and signing HelloWorld	24
Packaging via command line.....	25
Signing HelloWorld via command line.....	25
Signing HelloWorld via MOTODEV Studio for UIQ	25
Installing your application.....	27
Deploy via MOTODEV Studio for UIQ	27
References	30



Appendix A: Useful links	31
Registration Link	31
Download Links.....	31
UIQ SDK Download link:	31
Motorola PcSync download link:	31
Developer Resources	31
Appendix B: Troubleshooting	32
Compilation and build issue when building Symbian OS application	32
Unexpected Windows OS behavior after UIQ SDK installation/uninstall.....	32
GCCE target build fails with [*** missing separator. Stop] build errors.....	32
Signing issues	32
General signing guidelines.....	33
Appendix C: Self-Signing	34
Appendix D: PcSync installation and application deployment	35

An introduction to UIQ/Symbian development

This article is a supplement to educational material available via the MOTODEV developer site such as:

- “Building, Packaging, Signing, And Deploying From The Command Line”
- “QuickStart: MOTODEV Studio, MOTODEV Studio for UIQ”

Additional related material is provided by external technology partners such as UIQ, Symbian and Symbian Signed and are cited in the References section along with some useful links in Appendix A.

This article aims to provide you with a brief overview on how to develop a UIQ native application for Motorola UIQ devices. The focus is on two audiences, commercial developers and non-commercial developers. A commercial developer is one who intends to sell their Symbian signed native UIQ application. This requires a publisher ID, and normally, testing via the emulator and an actual device. A non-commercial developer may be a student, a hobbyist, or someone interested in developing for the UIQ/Symbian platform but not intending to have Symbian sign the application for commercial purposes or someone not having access to an actual device. We will only discuss the native application development process.

Background

Motorola UIQ devices are mobile phones based on the Symbian operating system (OS). The OS uses the UIQ Graphical User Interface (GUI) and development platform, and provides additional components to the Symbian OS. Both native and Java ME applications can be developed on this platform. Some well known Motorola UIQ devices based on the Symbian/UIQ platform are the A1000, M1000, MOTO Z8, and MOTO Z10.

Commercial development steps

- 1 Obtaining the tool and IDE
- 2 Building an application
- 3 Using the emulator
- 4 Packaging and signing an application
- 5 Installing the application

Table 1: Commercial developer track

Requirements
Subscriptions/Memberships: <ul style="list-style-type: none">▪ Publisher ID: www.trustcenter.de/en/index.htm▪ Symbian Signed Account: www.symbiansigned.com▪ UIQ Developer Account: http://www.developer.uiq.com/
Software: <ul style="list-style-type: none">▪ Windows 2000/ Windows XP▪ Motorola UIQ handset
Hardware: <ul style="list-style-type: none">▪ 1.1 GB required▪ UIQ SDK 3.1 or higher dependent-specific device/SDK requirements: www.symbiansigned.com
Optional software: <ul style="list-style-type: none">▪ MOTODEV Studio for UIQ: https://developer.motorola.com/docstools/motodevstudio/uiq/downloads/▪ Motorola PcSync: https://developer.motorola.com/docstools/motodevstudio/uiq/downloads/

Non-commercial development steps

- 1 Obtaining the tool and IDE
- 2 Building an application
- 3 Using the emulator



Table 2: Non-commercial developer track

Requirements
Subscriptions/Memberships: <ul style="list-style-type: none">▪ UIQ Developer Account: http://www.developer.uiq.com/
Software: <ul style="list-style-type: none">▪ Windows 2000/ Windows XP▪ UIQ SDK 3.1 or higher dependent-specific device/SDK requirements
Hardware: <ul style="list-style-type: none">▪ 1.1 GB required
Optional software: <ul style="list-style-type: none">▪ MOTODEV Studio for UIQ: https://developer.motorola.com/docstools/motodevstudio/uiq/downloads/

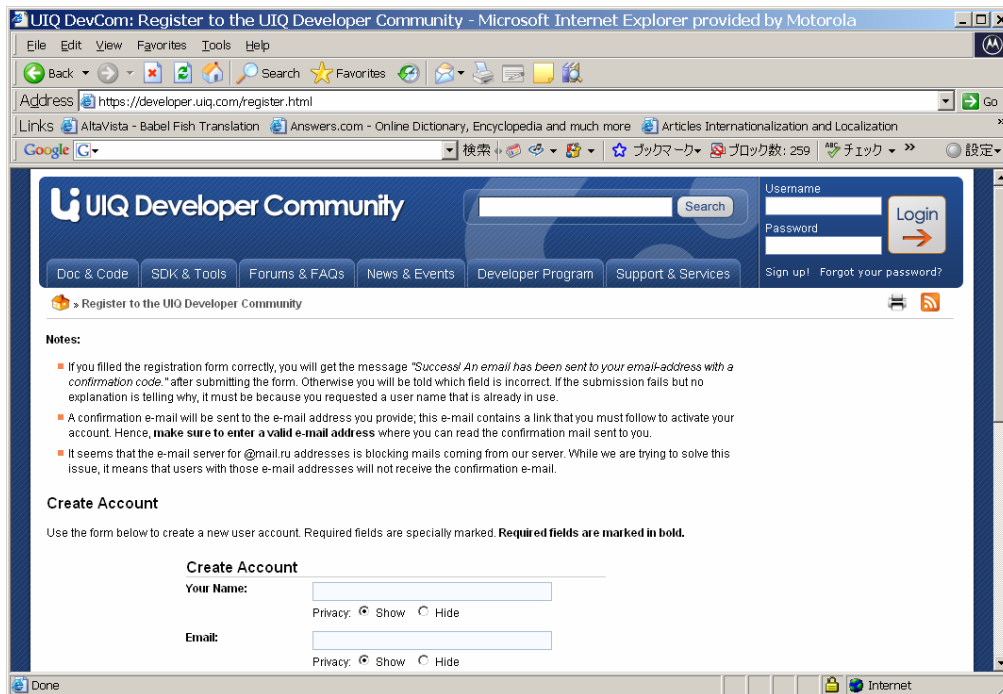
Developing UIQ applications

There are five steps to develop applications for UIQ.

- 1 Register at UIQ Developer Community
- 2 Download UIQ SDK
- 3 Install SDK
- 4 Download MOTODEV Studio for UIQ
- 5 Install and configure MOTODEV Studio for UIQ

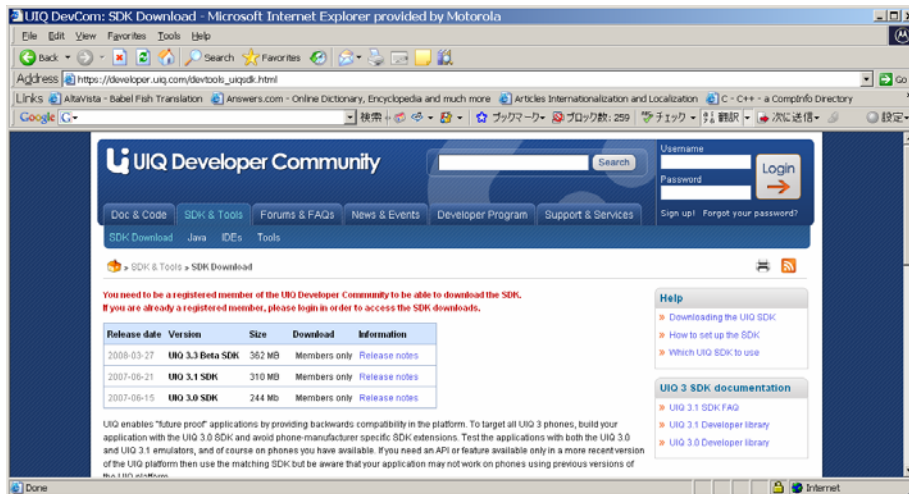
Register at UIQ Developer Community

Registration at the UIQ Developer Community site is required to download the UIQ SDK. Register at the following site: <https://developer.uiq.com/register.html>



Download UIQ SDK

Once you have registered, download the appropriate SDK at the following location: https://developer.uiq.com/devtools_uiqsdk.html



For the most current list of Motorola UIQ/Symbian devices, refer to the <https://developer.motorola.com/products/handsets/> page and filter on Operating System, selecting either UIQ or Symbian. Once you click to select a particular handset, the resulting product page lists the UIQ SDK version to download.

NOTE: MOTO Z8 supports the UIQ 3.1 SDK while MOTO Z10 supports the UIQ 3.2 SDK

Install SDK

The installation installs four components:

- Emulator
- C++ developer component
- Java ME component
- UIQ SDK documentation

Refer to the UIQ SDK release notes and/or ReadMe.txt for complete details regarding these components.

Once you have downloaded the UIQ SDK file, double click on the executable to start the installation and follow the prompts accordingly. We recommend that you select an empty or non-existing directory as the install location.

Once the SDK is installed, depending on the UIQ SDK version, you may be required to install the two compilers, both of which are included within the SDK installation package.

- GCC compiler (GCCE) for device builds
- X86 compiler (WINSWC) for emulator builds

Please refer to [UIQ FAQ Document 25](#) for more information regarding compiler installation. Release notes and known issues can be found in the ReadMe.txt typically located in the default path `Symbian > UIQversion` (where `<version>` refers to a specific SDK version name for example: UIQ3.1, or

UIQ3.3SDK_Beta) of the SDK installation directory. Release notes are also available from the link to the right of the UIQ SDK download link.

Download MOTODEV Studio for UIQ

To download MOTODEV Studio for UIQ, you need to log into the MOTODEV Developer portal site.

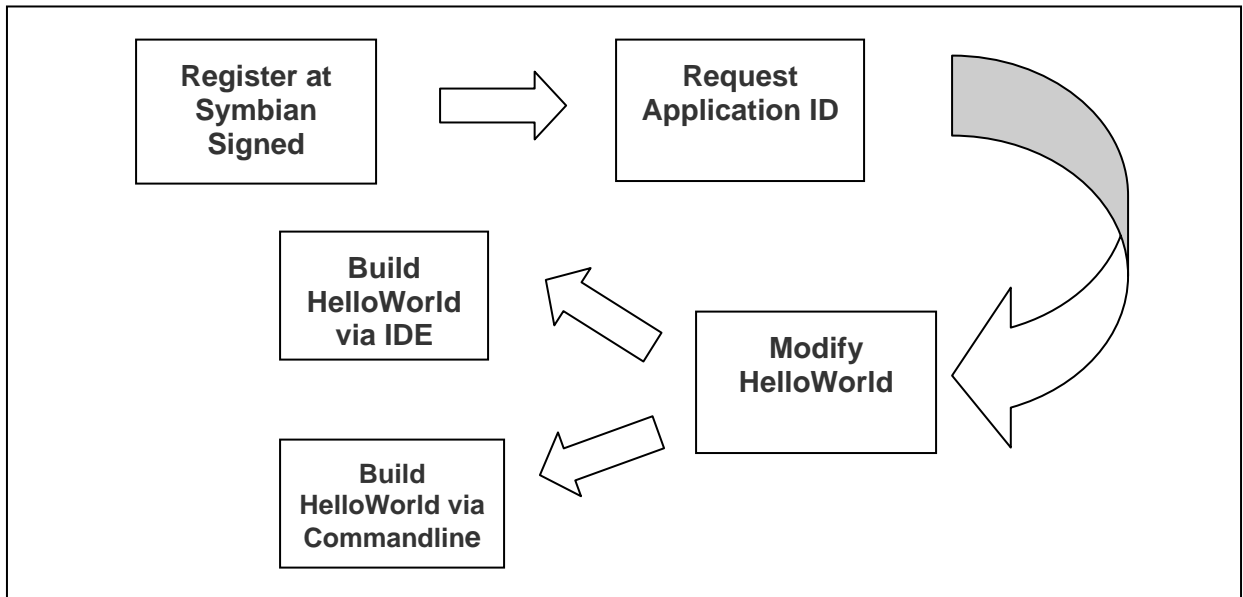
<http://developer.motorola.com/docstools/motodevstudio/uiq/>

Install MOTODEV Studio for UIQ

Double click the MOTODEV Studio for UIQ executable to start the installation process. Follow the installation wizard to install MOTODEV Studio for UIQ. Ensure that MOTODEV Studio for UIQ is installed on the same drive where you installed the UIQ SDK in the previous step.

Building a native application

Building a native application applies to both commercial and non-commercial developers.



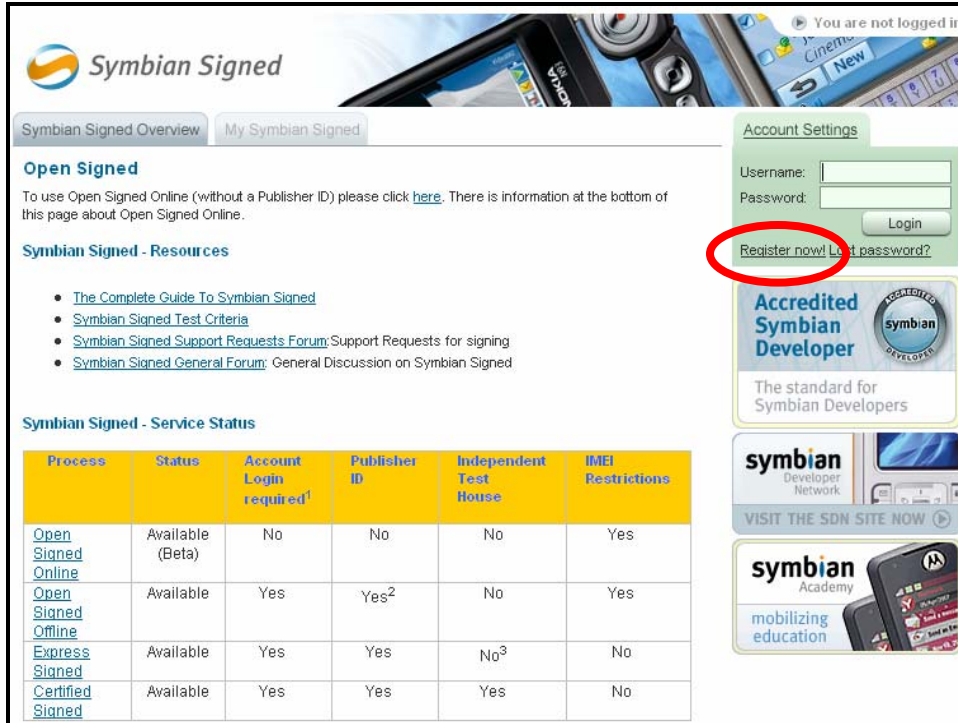
There are five steps to building a native UIQ application:

- 1 Register at Symbian Signed
- 2 Request application UID
- 3 Modify HelloWorld
- 4 Build HelloWorld via command line
- 5 Build HelloWorld via IDE

Register at Symbian Signed

This is a required step for commercial developers to request application Universal Identifiers (UIDs). Non-commercial developers are not required to follow this step. Non-commercial developers can use a test UID (i.e. in the range 0xE0000000 to 0xFFFFFFFF) which does not require a Symbian Signed account.

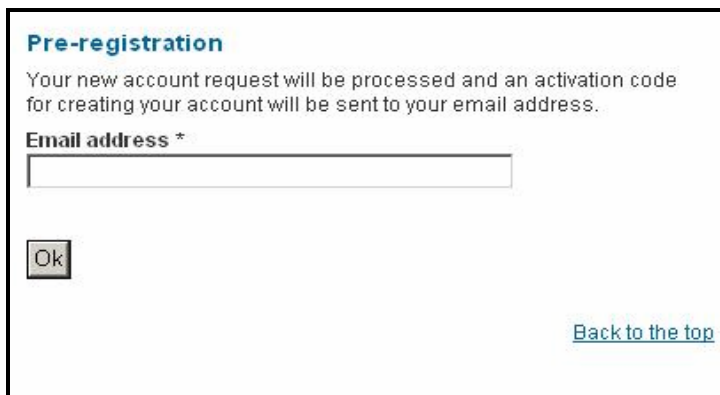
Open a browser and go to <https://www.symbiansigned.com/app/page>. Select the “Register now!” link, illustrated in the next figure circled in red.



The screenshot shows the Symbian Signed website interface. The main content area is titled "Open Signed" and includes a sub-section "Symbian Signed - Resources" with several links. A table titled "Symbian Signed - Service Status" is also present. On the right side, there is an "Account Settings" box with fields for "Username:" and "Password:", a "Login" button, and a "Register now! Lost password?" link, which is circled in red.

Process	Status	Account Login required ¹	Publisher ID	Independent Test House	IMEI Restrictions
Open Signed Online	Available (Beta)	No	No	No	Yes
Open Signed Offline	Available	Yes	Yes ²	No	Yes
Express Signed	Available	Yes	Yes	No ³	No
Certified Signed	Available	Yes	Yes	Yes	No

Follow the on-line form to pre-register for your Symbian Signed account by entering a valid email address and selecting the Ok button.

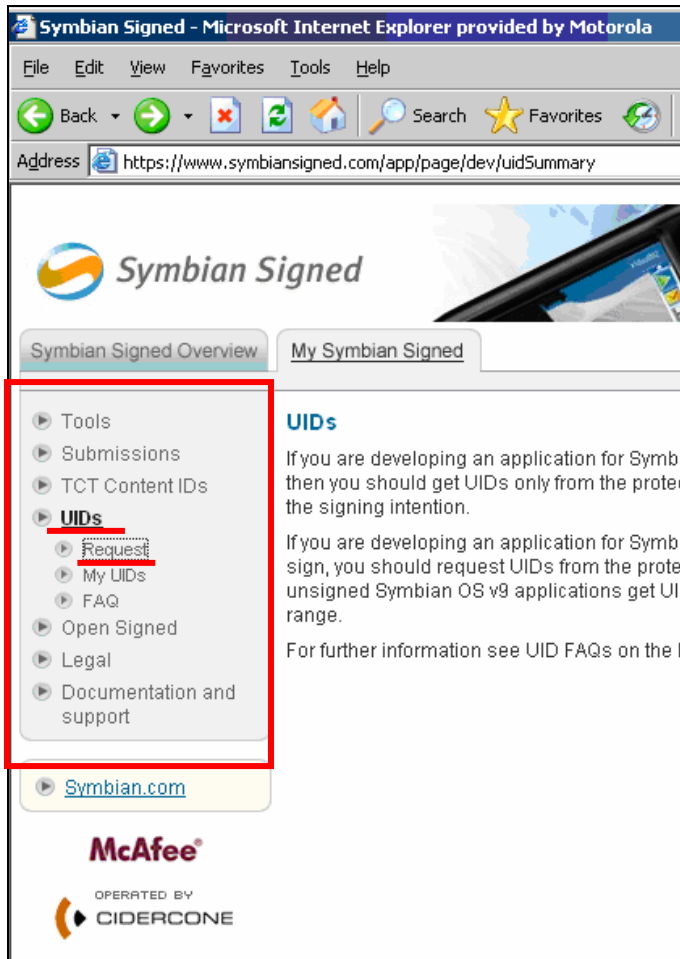


The screenshot shows a "Pre-registration" form. It includes a heading "Pre-registration", a paragraph explaining that an activation code will be sent to the email address, and a text input field labeled "Email address *". Below the input field is an "Ok" button and a "Back to the top" link.

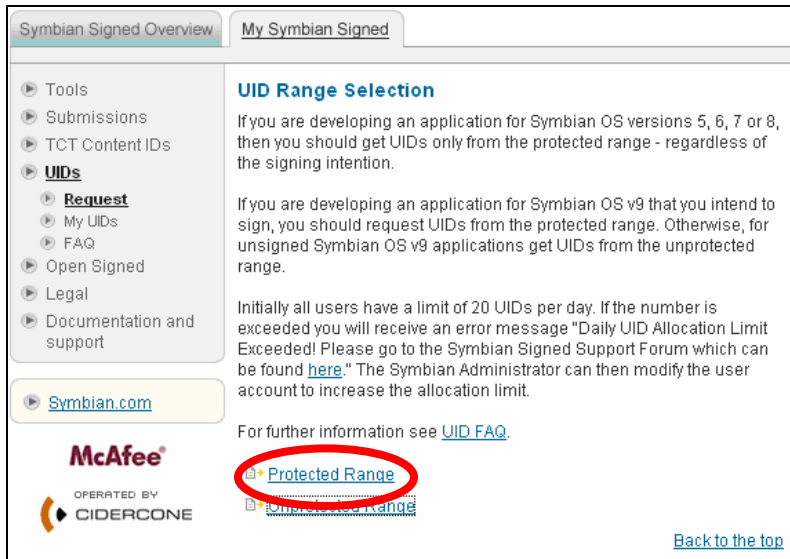
A confirmation email for pre-registration will be sent to the email account entered in the above form. At the time of this writing, the email contains a link to the Symbian Signed site to complete the registration process. Enter the required information into the form and submit your request.

Request application UID

This is the required step for commercial developers. Login to the Symbian Signed site and select the [UIDs] link then the [Request] link as shown below.



Since the development is targeting Symbian OS 9.x, select the [Protected Range] link.



Symbian Signed Overview | My Symbian Signed

- Tools
- Submissions
- TCT Content IDs
- UIDs**
 - Request**
 - My UIDs
 - FAQ
- Open Signed
- Legal
- Documentation and support

[Symbian.com](#)

McAfee
OPERATED BY
CIDERONE

UID Range Selection

If you are developing an application for Symbian OS versions 5, 6, 7 or 8, then you should get UIDs only from the protected range - regardless of the signing intention.

If you are developing an application for Symbian OS v9 that you intend to sign, you should request UIDs from the protected range. Otherwise, for unsigned Symbian OS v9 applications get UIDs from the unprotected range.

Initially all users have a limit of 20 UIDs per day. If the number is exceeded you will receive an error message "Daily UID Allocation Limit Exceeded! Please go to the Symbian Signed Support Forum which can be found [here](#)." The Symbian Administrator can then modify the user account to increase the allocation limit.

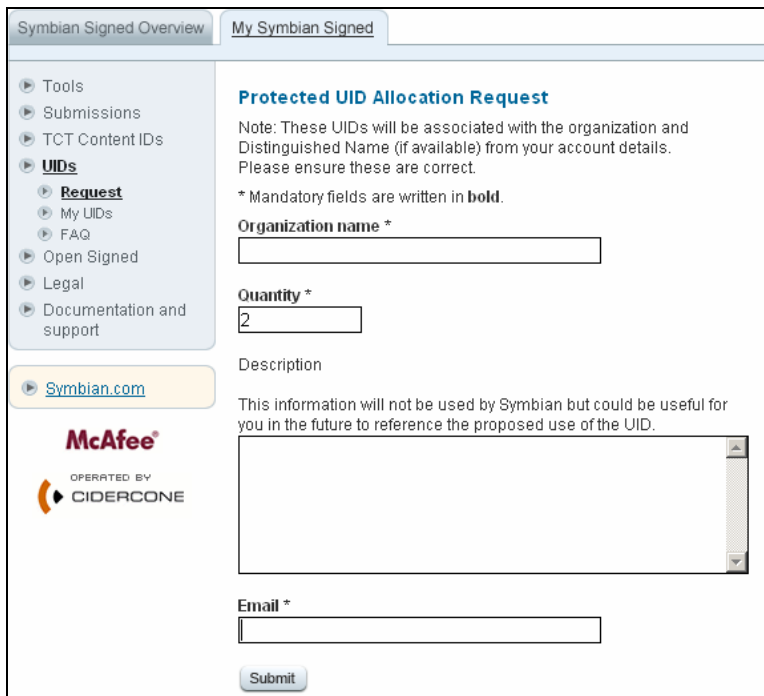
For further information see [UID FAQ](#).

[Protected Range](#)

[Unprotected Range](#)

[Back to the top](#)

Fill in the appropriate information for all mandatory fields and select the Submit button to receive your UIDs. Developers are still able to use UIDs from the Unprotected range during test phase, but ultimately, commercial developers need to use UIDs from the Protected range for a Symbian Signed application and need to follow these steps.



Symbian Signed Overview | My Symbian Signed

- Tools
- Submissions
- TCT Content IDs
- UIDs**
 - Request**
 - My UIDs
 - FAQ
- Open Signed
- Legal
- Documentation and support

[Symbian.com](#)

McAfee
OPERATED BY
CIDERONE

Protected UID Allocation Request

Note: These UIDs will be associated with the organization and Distinguished Name (if available) from your account details. Please ensure these are correct.

* Mandatory fields are written in **bold**.

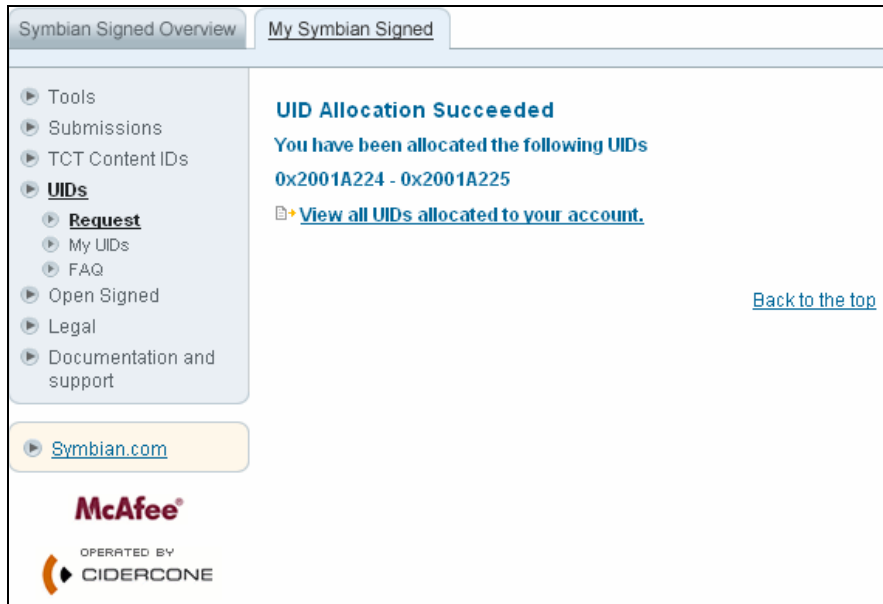
Organization name *

Quantity *

Description
This information will not be used by Symbian but could be useful for you in the future to reference the proposed use of the UID.

Email *

You will receive your UID, or UID range, directly from the Symbian Signed website when the page refreshes.



The screenshot shows a web interface for Symbian Signed. At the top, there are two tabs: 'Symbian Signed Overview' and 'My Symbian Signed'. The 'My Symbian Signed' tab is active. On the left side, there is a navigation menu with the following items: Tools, Submissions, TCT Content IDs, **UIDs** (expanded), Request, My UIDs, FAQ, Open Signed, Legal, and Documentation and support. Below the menu is a button for 'Symbian.com'. At the bottom left, there is a McAfee logo and the text 'OPERATED BY CIDER CONE'. The main content area displays a success message: 'UID Allocation Succeeded' in blue, followed by 'You have been allocated the following UIDs' and the specific UID range '0x2001A224 - 0x2001A225'. Below this, there is a link to 'View all UIDs allocated to your account.' and a 'Back to the top' link.

Modify HelloWorld

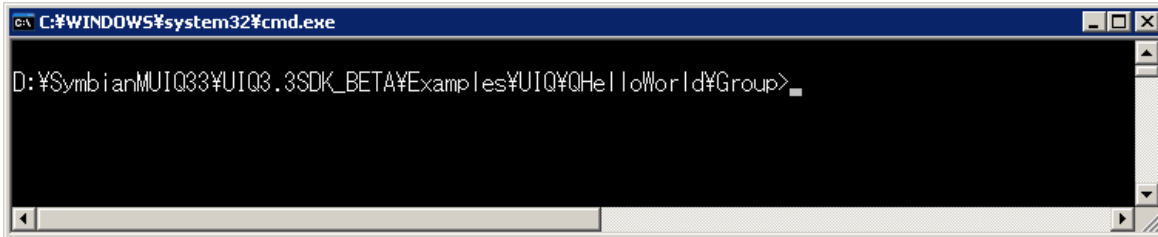
The QHelloWorld application uses a project identifier UID value of 0xE0001001 which is within the test range. Commercial developers can follow this step to modify the QHelloWorld sample application UID from this test range value to a UID value they received through Symbian Signed. For example as shown above, the UID 0x2001A224 can be used in place of 0xE0001001.

The QHelloWorld application is available in the <UIQ SDK>/Examples/UIQ/QhelloWorld directory after SDK installation and can be used as the basis of an application skeleton to create your own unique application. In QHelloWorld, the test UID is used in four files. Replace the test UID with one of the requested UIDs in these files:

```
<UIQ SDK>/Examples/UIQ/QhelloWorld/Group/HelloWorld.mmp  
<UIQ SDK>/Examples/UIQ/QhelloWorld/Inc/HelloWorldUid.h  
<UIQ SDK>/Examples/UIQ/QhelloWorld/Group/SIS/HelloWorld_winscw.pkg  
<UIQ SDK>/Examples/UIQ/QhelloWorld/Group/SIS/HelloWorld_gcce.pkg
```

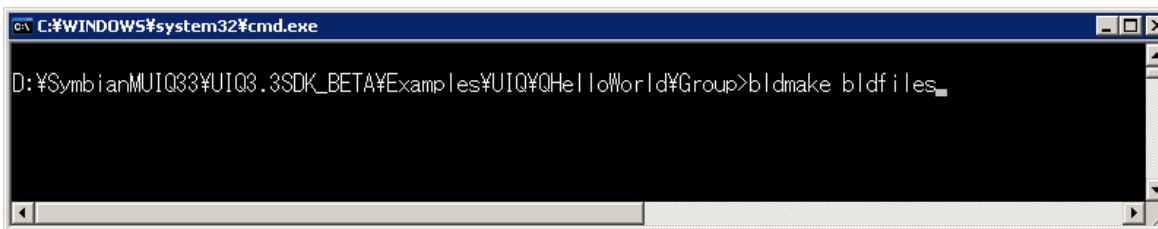
Build HelloWorld via command line

Open a command prompt and navigate to the Examples\UIQ\QHelloWorld\Group within your UIQ SDK installation directory. The following are commands necessary to build the application and a brief explanation of each. See the UIQ SDK documentation for more details.



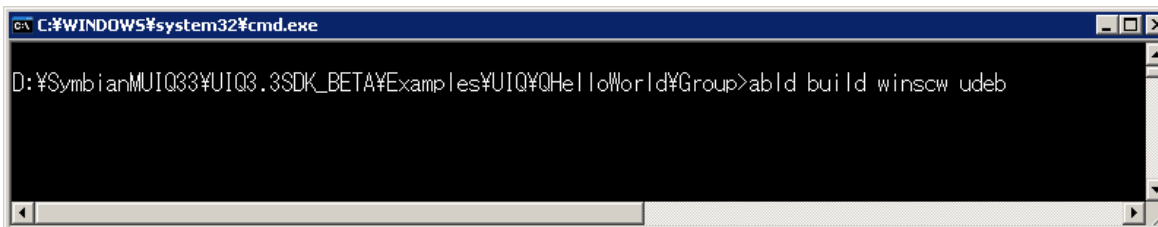
```
C:\WINDOWS\system32\cmd.exe
D:\SymbianMUIQ33\UIQ3.3SDK_BETA\Examples\UIQ\HelloWorld\Group>
```

The `bldmake bldfiles` command processes the component description file (`bld.inf`) and creates an `ABLD.BAT` file within the same directory as the component description file.



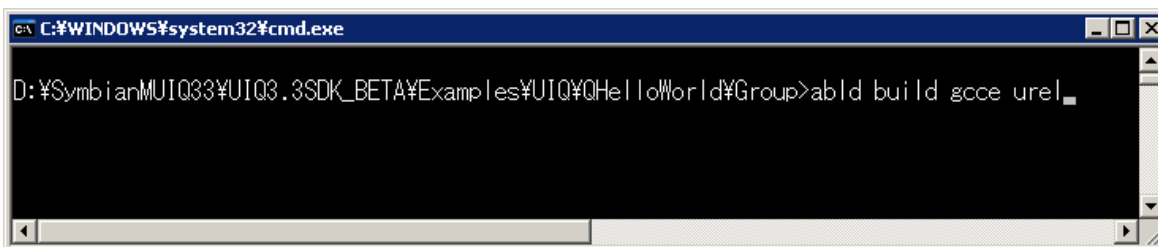
```
C:\WINDOWS\system32\cmd.exe
D:\SymbianMUIQ33\UIQ3.3SDK_BETA\Examples\UIQ\HelloWorld\Group>bldmake bldfiles
```

The emulator build command `abld build <TARGET> <BUILD TYPE>` `abld build winscw udeb` builds a debug version that can be run and tested in the emulator.



```
C:\WINDOWS\system32\cmd.exe
D:\SymbianMUIQ33\UIQ3.3SDK_BETA\Examples\UIQ\HelloWorld\Group>abld build winscw udeb
```

The Device build command `abld build gcce urel` builds a device release version that can be installed on the handset.

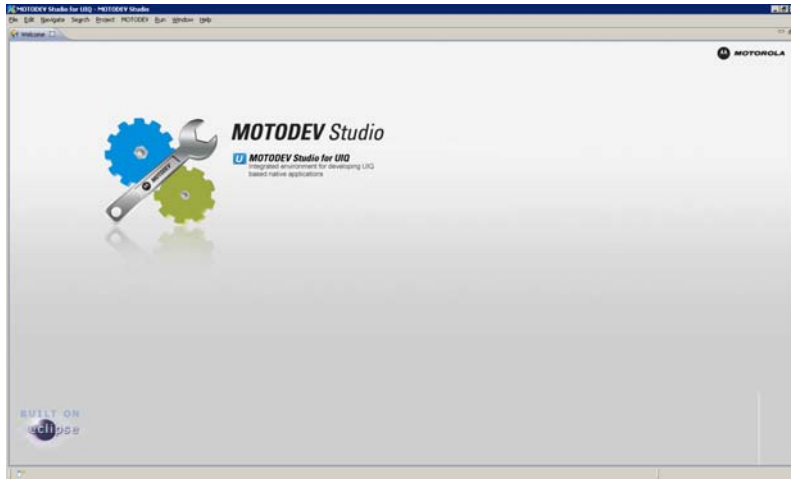


```
C:\WINDOWS\system32\cmd.exe
D:\SymbianMUIQ33\UIQ3.3SDK_BETA\Examples\UIQ\HelloWorld\Group>abld build gcce urel
```

Using MOTODEV Studio for UIQ to build HelloWorld

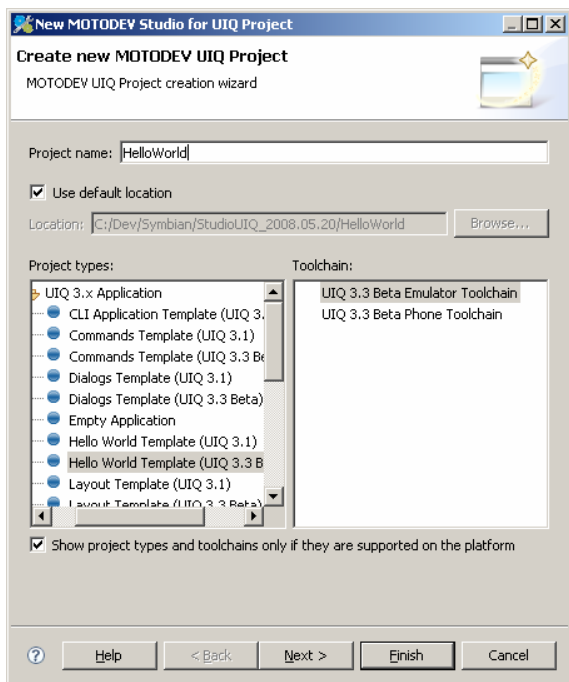
MOTODEV Studio for UIQ is the recommended Integrated Development Environment (IDE) for development on the UIQ platform. The IDE also includes project templates that you can modify quickly to develop your applications. See the section ***Steps to develop UIQ applications*** earlier in this article for information on obtaining and installing MOTODEV Studio for UIQ.

Start MOTODEV Studio for UIQ: Start > Programs > MOTODEV Studio > MOTODEV Studio for UIQ v0.5.0 > MOTODEV Studio for UIQ v0.5.0.

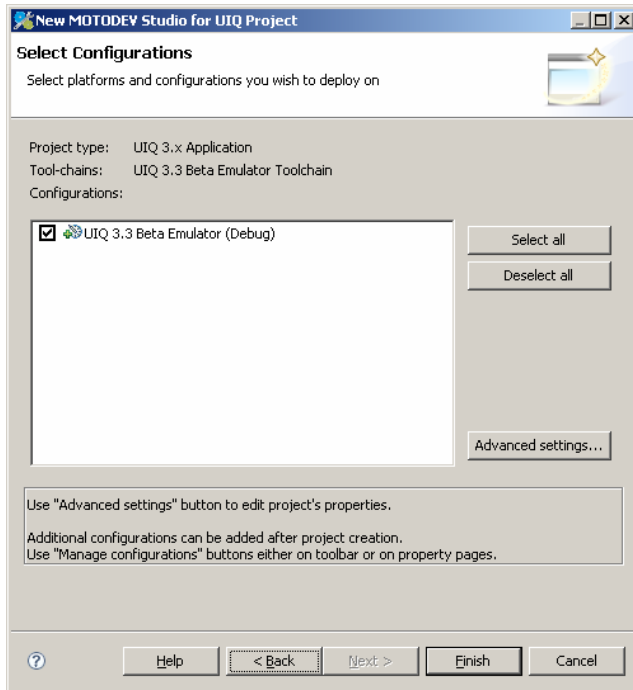


To create a HelloWorld application using the included template, select the following from the MOTODEV menu: MOTODEV > Studio for UIQ > Create new project

- 1 From the Create new MOTODEV UIQ Project dialog, enter a meaningful name for the project in the Project name textbox (HelloWorld is used in this example).
- 2 Expand the UIQ 3.x Application directory in the Project types pane and select the Hello World Template (UIQ 3.3 Beta) project.

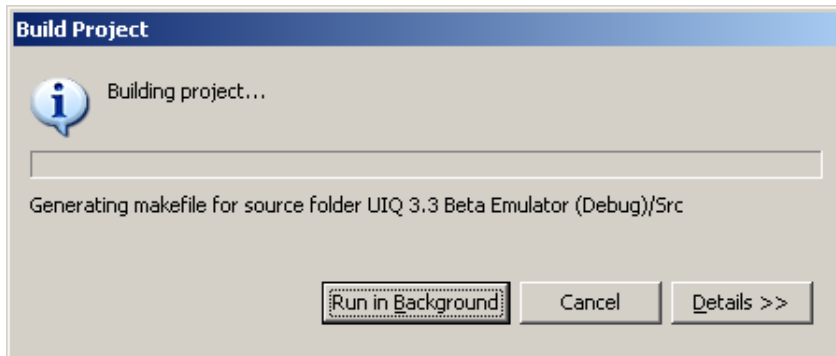


- 3 Select UIQ 3.3 Beta Emulator Toolchain from the Toolchain pane.
- 4 Select the [Next>] button to view/modify advanced settings for the project or select [Finish] if you are done.

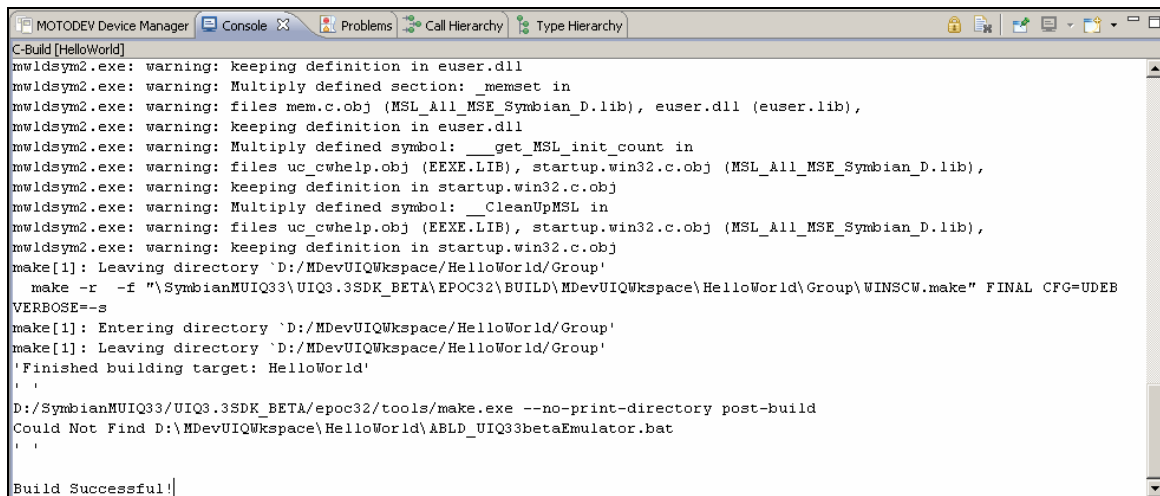


See the Help > Help Contents menu for further explanation of the settings available from the [Advanced settings...] button. Many of these properties are still available to the developers after project creation from the Project > Properties menu.

- 5 With the project selected in the Project Explorer pane, select Project > Build Project menu.



The Console Tab displays build information and should complete with a 'Build Successful!' message. See Appendix B: Troubleshooting for hints on fixing any issues that arise at this point.



```
C-Build [HelloWorld]
mwlidsym2.exe: warning: keeping definition in euser.dll
mwlidsym2.exe: warning: Multiply defined section: _memset in
mwlidsym2.exe: warning: files mem.c.obj (MSL_All_MSE_Symbian_D.lib), euser.dll (euser.lib),
mwlidsym2.exe: warning: keeping definition in euser.dll
mwlidsym2.exe: warning: Multiply defined symbol: __get_MSL_init_count in
mwlidsym2.exe: warning: files uc_cwhelp.obj (EEXE.LIB), startup.win32.c.obj (MSL_All_MSE_Symbian_D.lib),
mwlidsym2.exe: warning: keeping definition in startup.win32.c.obj
mwlidsym2.exe: warning: Multiply defined symbol: __CleanupMSL in
mwlidsym2.exe: warning: files uc_cwhelp.obj (EEXE.LIB), startup.win32.c.obj (MSL_All_MSE_Symbian_D.lib),
mwlidsym2.exe: warning: keeping definition in startup.win32.c.obj
make[1]: Leaving directory `D:/MDevUIQWkworkspace/HelloWorld/Group'
  make -r -f "\SymbianMUIQ33\UIQ3.3SDK_BETA\EPOC32\BUILD\MDevUIQWkworkspace\HelloWorld\Group\WINSW.make" FINAL CFG=UDEB
VERBOSE=-s
make[1]: Entering directory `D:/MDevUIQWkworkspace/HelloWorld/Group'
make[1]: Leaving directory `D:/MDevUIQWkworkspace/HelloWorld/Group'
'Finished building target: HelloWorld'
'
'
D:/SymbianMUIQ33/UIQ3.3SDK_BETA/epoc32/tools/make.exe --no-print-directory post-build
Could Not Find D:\MDevUIQWkworkspace\HelloWorld\&BLD_UIQ33betaEmulator.bat
'
'
Build Successful!
```

- 6 Developers can substitute their own UIDs, as stated in the building from command line section, by modifying the following files in the project and then rebuilding the Project as in step 5 above.

Inc/HelloWorldUid.h

Group/HelloWorld.mmp

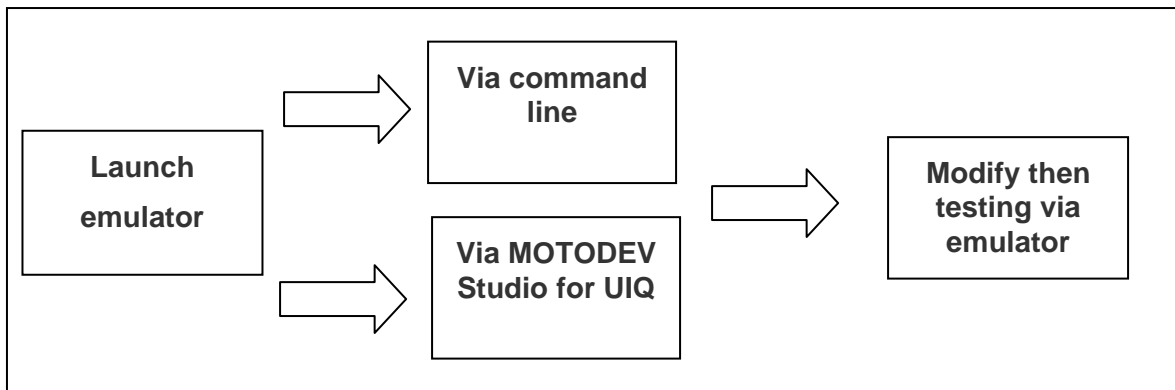
Group/SIS/HelloWorld_gcce.pkg

Group/SIS/HelloWorld_winscw.pkg

Using the emulator

The emulator can be used by both commercial and non-commercial developers. Here are the steps for using the emulator:

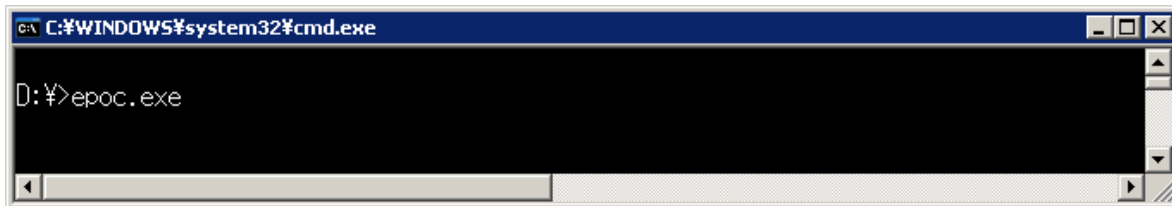
- 1 Launch emulator
- 2 Via command line or via MOTODEV Studio for UIQ
- 3 Modify and test HelloWorld via the emulator



Launching HelloWorld using the command line

Start a DOS window from Start > Programs > Accessories > Command Prompt

Executing the 'epoc' command launches the emulator.

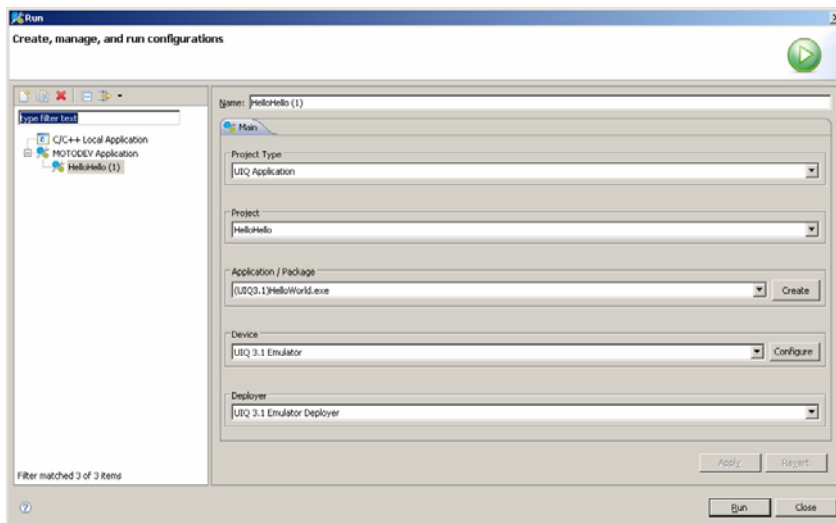


The emulator can be operated using your PC mouse key presses. You can also activate the scroll bar on the right hand side of the display using your PC mouse. Use your mouse to navigate to the Application Launcher once the emulator has started up. Select the Hello World program by clicking on the four way navigation keys then the FIRE/SELECT key to launch the Hello World application.



Launching HelloWorld using the IDE

Select [Run] > [Run As] > [MOTODEV Application] to open the Run dialog. Select the [Run] button to launch the application in the emulator. The emulator launched from the IDE with these steps is the same emulator as the one launched from the command line with the `epoc` command, so developers can operate the emulator in the same way to test the HelloWorld application.



Modifying and testing the HelloWorld application

The final part in this build section covers a modification in the application, rebuilding the project, and verifying the modification in the emulator launched from the IDE. For this demonstration we will modify the name that displays in the Application launcher screen.

- Open the `Reg/HelloWorld_loc_rls` file in the Project Explorer window by double-clicking the file or by dragging and dropping the file from Project Explorer into the workspace.
- Modify the `String_r_helloworld_caption` value from "HelloWorld" to "My Project" or some other string of your choice. (The double quotes are necessary).
- Save the change by selecting the File > Save menu.
- Rebuild the project from the Project > Build Project menu.
- Launch the emulator to verify the modification using the Run > Open Run Dialog... menu.
- Select [Run] to launch emulator and verify that the modification displays in the emulator Application Launcher.



Signing and packaging an application

Here are the steps for commercial developers to package and sign an application:

- 1 Signing overview
- 2 Signing capabilities
- 3 Signing options
- 4 Packaging and signing HelloWorld

Signing overview

Signing is the process of using a tamper-proof digital certificate to encode an application. In order to access the “Capability protected” APIs in a Symbian OS application which are declared at build-time, a digital certificate is required. A capability is a command statement included in a native application's MMP build file and/or a token used to allow access to a protected resource. For complete details on regarding “Symbian Signed” visit the portal site (<https://www.symbiansigned.com>). This article briefly describes four signing options available to developers. As an alternative, applications that only require platform security capabilities belonging to the group “user” or “basic” capabilities can use the self-signing option described in Appendix C.

Signing capabilities

The Symbian OS Platform Security Model defines 20 capabilities. The mechanism used to grant capabilities to applications is via the signing process. There are four groupings of capability types.

User Capabilities – User capabilities are available through all signing options. They are:

- LocalServices
- Location
- NetworkServices
- ReadUserData
- UserEnvironment
- WriteUserData

System Capabilities – All system capabilities, including restricted, are available through Open Signed (with a Publisher ID) and Certified Signed options. They are:

- PowerMgmt
- ProtServ
- ReadDeviceData
- SurroundingsDD
- SwEvent
- TrustedUI

- WriteDeviceData.

Restricted Capabilities – Express Signed does not allow access to restricted capabilities (CommDD, DiskAdmin, NetworkControl, and MultimediaDD).

Device Manufacturer Capabilities - Symbian Signed refers to the most sensitive capabilities, specifically AllFiles, DRM, and TCB, as Device Manufacturer Capabilities. These are only available through the Open Signed (with a Publisher ID) and Certified Signed options and require Device Manufacturer approval.

Signing options

Capabilities are listed in an application's MMP file depending on the API being accessed. The signing options available to developers are mentioned here and described in the next section.

Open signed

Open signed provides two signing methods which makes it easy for developers to sign applications for limited deployment to known devices, either for testing or for personal use. Deployment is restricted by the device's International Mobile Equipment Identity (IMEI).

Open signed online

Open signed online has these features:

- Simple online process
- Publisher ID not required
- Symbian Signed account not required
- Tool downloads not required, so it is platform independent
- All User and System Capabilities may be requested (excludes the Restricted Capabilities and Device Manufacturer Capabilities)
- Installation of applications is restricted to one device, specified by the IMEI
- No cost to developers
- UIDs from the protected range must have been allocated to the email address associated with Symbian Signed account
- UIDs from the "development range" may be used

Open signed offline

Open signed offline has these features:

- Publisher ID is required
- Symbian Signed account is required
- All user, system and restricted capabilities may be requested
- Applications are restricted by IMEI, but up to 1000 devices may be specified

- The request process generates a Developer Certificate which is downloaded and used for local signing of applications
- Developer Certificates can be used to sign multiple Symbian Installation Script (SIS) files.

Express signed

One attractive characteristic of this signing option for developers wishing to reduce development costs is that no independent testing is required. However, applications are audited for compliance.

- Publisher ID is required
- Symbian Signed account is required
- User and System Capabilities may be requested (Restricted Capabilities cannot be requested)
- Unrestricted application deployment
- Signed applications are valid for ten years from the date of signing
- Single applications and multiple batched applications may be submitted
- Independent testing is not required, but applications must meet the Symbian Signed Test Criteria, and must have been tested on that basis before submission

Certified signed

This option is recommended for commercial developers targeting Motorola UIQ devices. It provides access to all but Device Manufacturer Capabilities, but requires applications to be submitted for independent testing. Certified Signed applications are entitled to use the 'for Symbian OS' logo to add differentiation and brand building. The main features are:

- Publisher ID is required
- Symbian Signed account is required
- Applications must be submitted to a Symbian Signed accredited test house as part of the signing process
- All User, System, and Restricted capabilities may be requested
- Access to Device Manufacturer capabilities requires manufacturer approval
- Signed applications are valid for ten years from the date of signing
- Single applications and multiple batched applications may be submitted
- Unrestricted application deployment
- A "Fast Track" option is available for Certified Signed

Packaging and signing HelloWorld

Packaging involves creating an installation file (.SIS) based on the statements included in a package (PKG) file. Two methods of packaging the HelloWorld application are demonstrated below. These steps assume that a developer certificate and key have already been obtained. For more detailed instructions, see the, "*A Guide to Symbian Signed*" document available on the Symbian Signed site.

Packaging via command line

Open a Command prompt and navigate to the QHelloWorld UIQ example located in your <UIQ SDK > \Examples \UIQ directory. These steps assume the application has already been built. The result of executing the `makesis` command is an unsigned .SIS file.

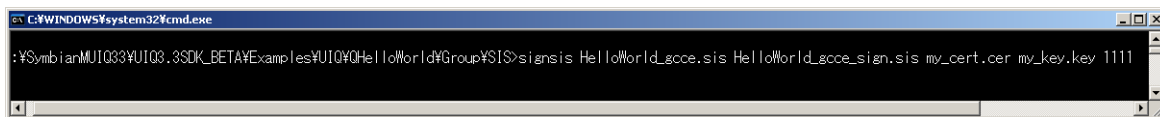


```
C:\WINDOWS\system32\cmd.exe
D:\SymbianMUIQ33\UIQ3.3SDK_BETA\Examples\UIQ\QHelloWorld\Group\SIS>makesis HelloWorld_gcce.pkg_
```

Signing HelloWorld via command line

Execute the `signsis` command to sign the application and get a signed SIS file.

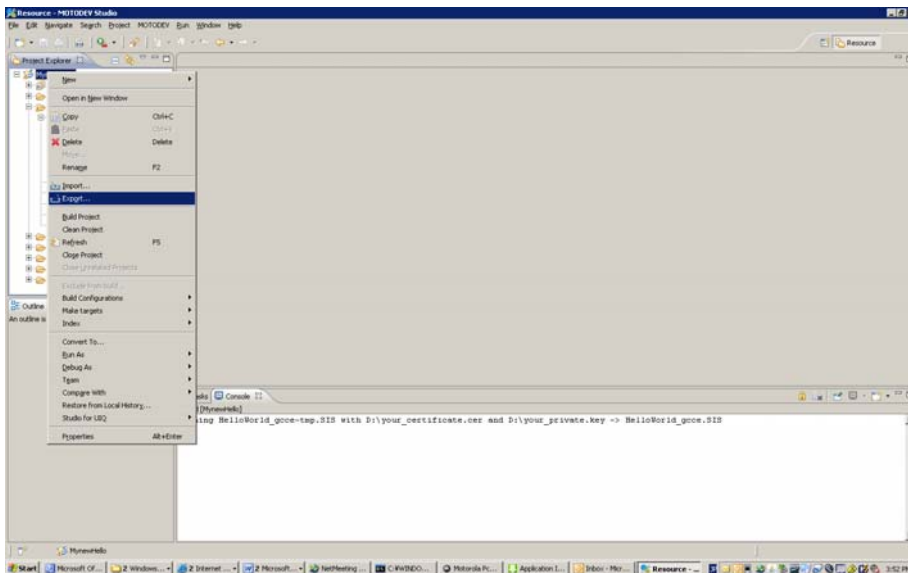
```
signsis <unsigned_sis_file_name.sis> <signed_sis_file_name.sis>
<certificate_name.cer> <key_name.key> <password>
```



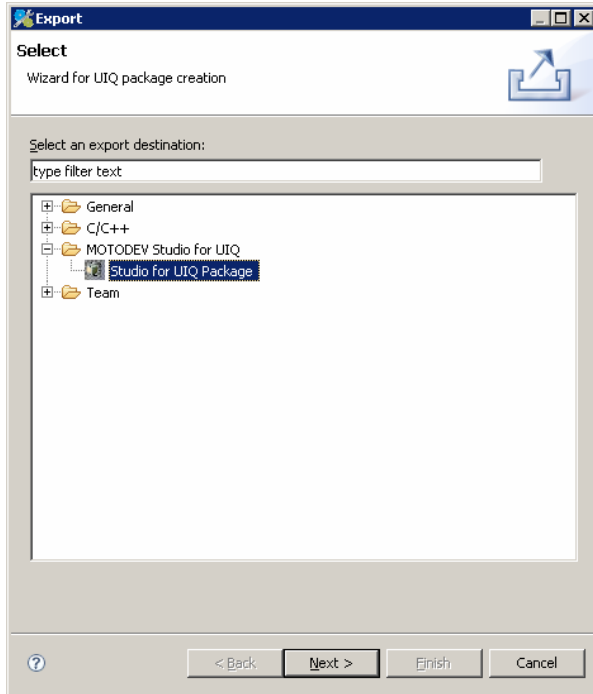
```
C:\WINDOWS\system32\cmd.exe
D:\SymbianMUIQ33\UIQ3.3SDK_BETA\Examples\UIQ\QHelloWorld\Group\SIS>signsis HelloWorld_gcce.sis HelloWorld_gcce_sign.sis my_cert.cer my_key.key 1111
```

Signing HelloWorld via MOTODEV Studio for UIQ

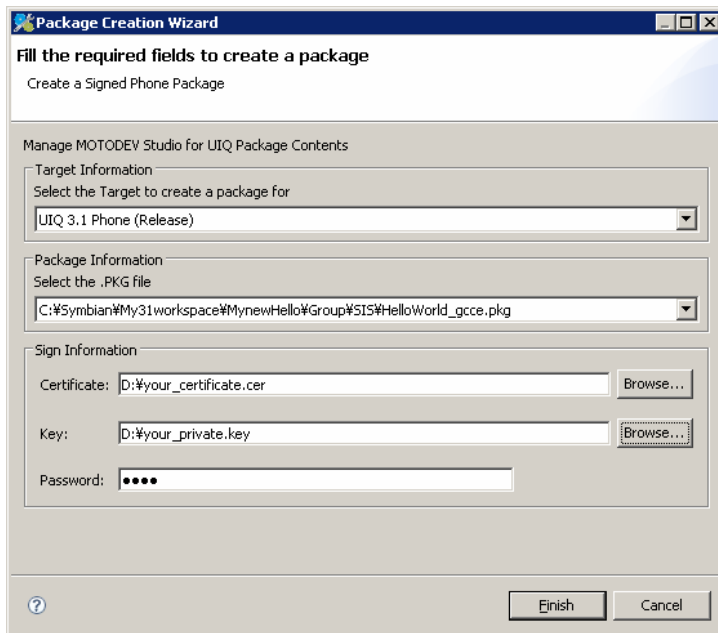
After building a GCCE Device target version of your application you are ready to package and sign for deployment to the target device. Connect your device to a PC using a USB cable or Bluetooth wireless connection. Right-click on the project name in Project Explorer pane and select [Export...] as shown below.



Select [Studio for UIQ Package] from Export applet as shown below.



Select the [Browse...] buttons and select your certificate and your key files and enter the password associated with them as shown here.



Select [Finish] to create the signed .SIS file for deployment to the target device.

Installing your application

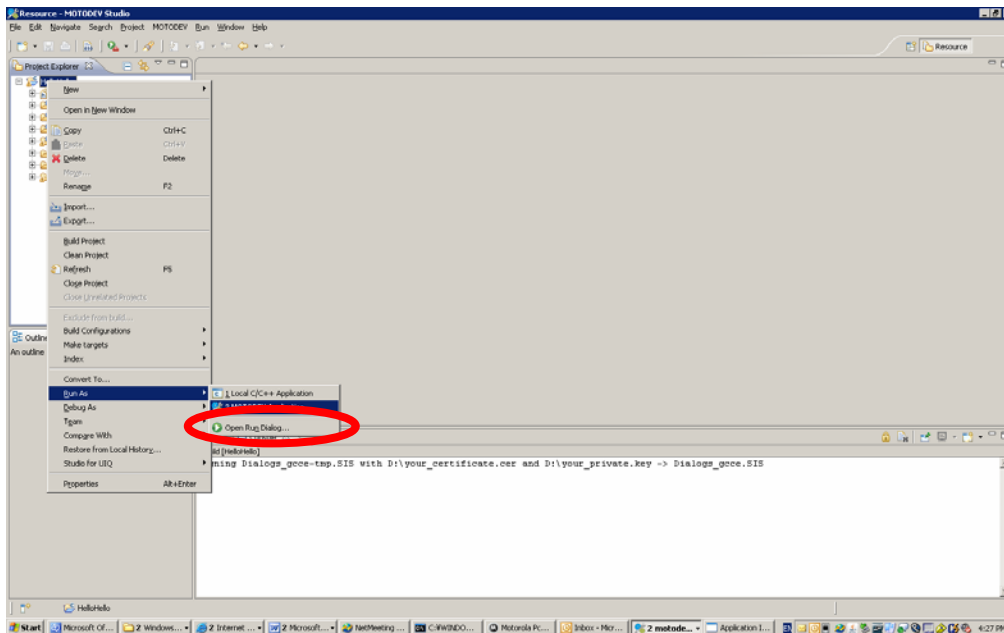
Commercial developers wanting to deploy their application need to complete the following steps:

- Deploy via MOTODEV Studio for UIQ
- [Obtain Motorola PcSync](#)
- Install PcSync
- Install HelloWorld

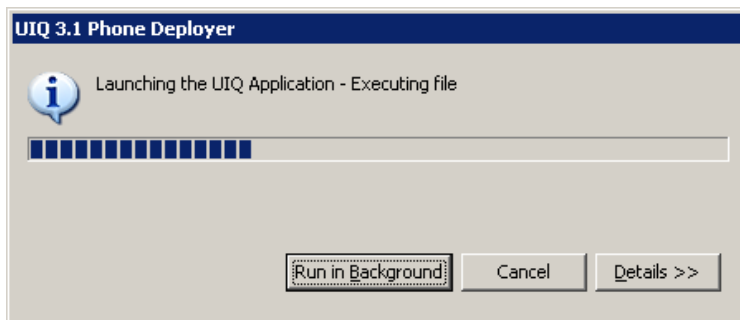
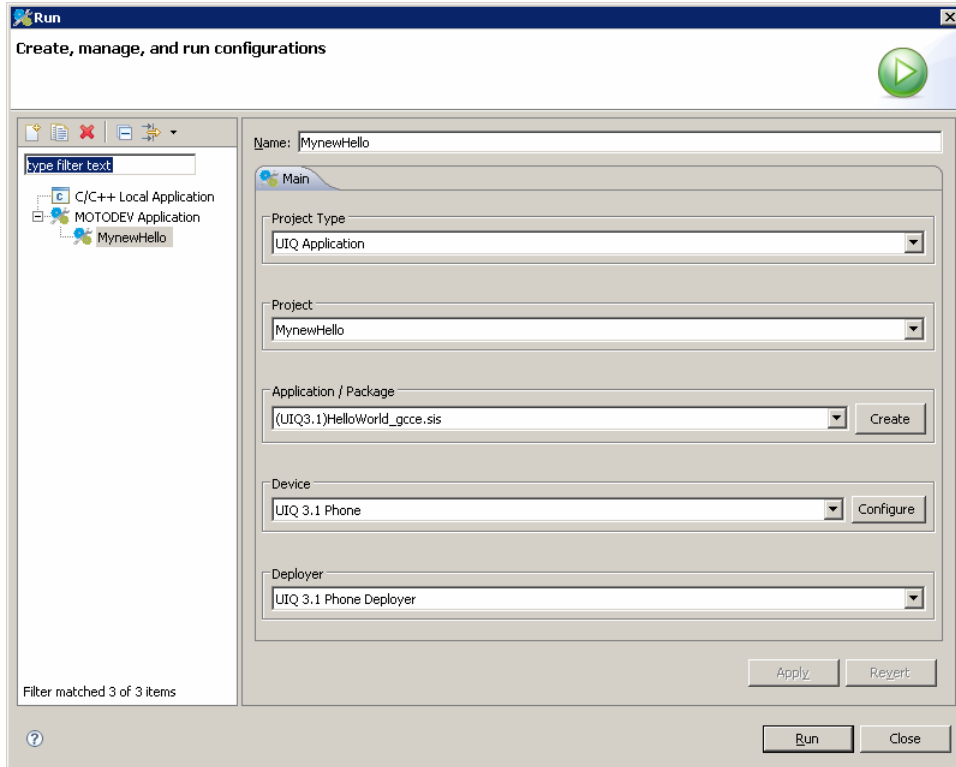
Deploy via MOTODEV Studio for UIQ

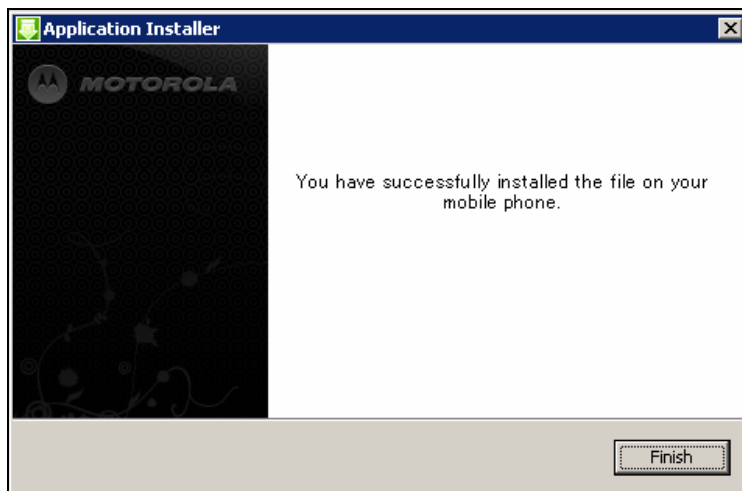
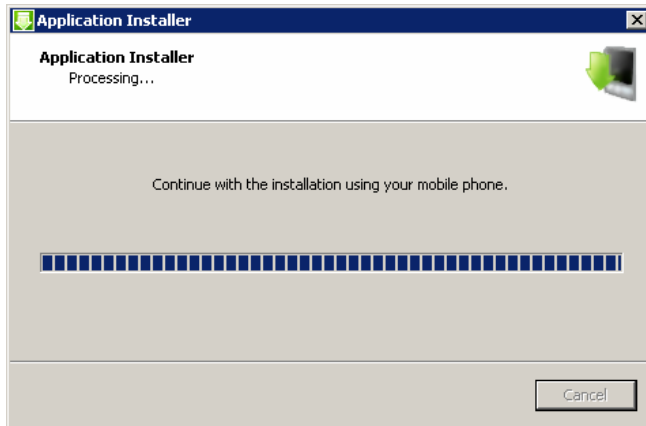
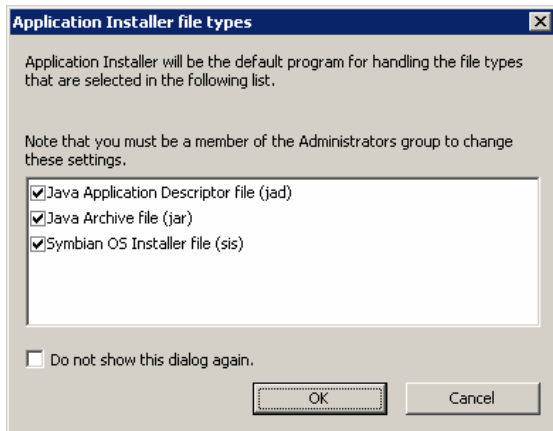
Connect your PC to the target device using USB cable. When using a Bluetooth wireless connection method, ensure you are properly connected with the target device using the PcSync *'How to connect wizard'*. Alternatively, PcSync can be used directly for deployment. See Appendix D for PcSync installation and application deployment.

Right-click on project in the Project Explorer pane and select Run As... > Open Run Dialog menu.



Select [Run] to deploy the application to target device.





References

A Guide to Symbian Signed' Symbian Software Limited

UIQ 3.1 Developer Library - http://developer.uiq.com/devlib/uiq_31/SDKDocumentation/doc_source/

UIQ 3.3 Developer Library

UIQ Developer Community - <http://www.uiqdeveloper.com/>

Symbian OS Library, Version 9.2

Symbian OS Library, Version 9.3

MOTODEV Developer Portal site: <http://developer.motorola.com/>

MOTO Z8 Device page - <https://developer.motorola.com/products/handsets/motoz8/>

MOTO Z10 Device page - <https://developer.motorola.com/products/handsets/motoz10/>



Appendix A: Useful links

Registration Link

<https://developer.uiq.com/register.html>

Download Links

UIQ SDK Download link:

https://developer.uiq.com/devtools_uiqsdk.html

Motorola PcSync download link:

direct.motorola.com/motography/pccd_vista_32.zip

Developer Resources

UIQ Developer Community, Forums & FAQs

<http://developer.uiq.com/ask.html>

FAQ 25 Setting up the SDK tools and compilers.

<http://developer.uiq.com/forum/kbclick.jspa?categoryID=18&externalID=25&searchID=314207>

Symbian Signed

<http://symbiansigned.com/>

Symbian Developer Network

<http://developer.symbian.com/main/index.jsp>

Appendix B: Troubleshooting

Compilation and build issue when building Symbian OS application

If you experience problems building for WINSCW target using the template in MOTODEV Studio for UIQ, please refer to the following UIQ SDK FAQ to verify your compiler path configuration.

[25] Setting up the SDK tools and compilers.

<https://developer.uiq.com/forum/kbclick.jspx?categoryID=18&externalID=25&searchID=321450>

Unexpected Windows OS behavior after UIQ SDK installation/uninstall

In very rare cases installing the SDK causes a loss of the environment paths. To help you restore your paths check <installdir>/path.log or save your path environment to desktop before installation. Restore your environment path to original if you experience any unexpected behavior due to missing environment path.

GCCE target build fails with [missing separator. Stop] build errors***

Verify that the C:\Program Files\CSL Arm Toolchain\bin; path is included in your environment path statement and that it is not enclosed in quotes.

Incorrect: <path statement>;"C:\Program Files\CSL Arm Toolchain\bin;"

Correct: <path statement>;C:\Program Files\CSL Arm Toolchain\bin;

Signing issues

- 1 "Security Information: Component failed security check. Cannot install."

This information message appears when trying to install an unsigned native Symbian application to the Motorola UIQ device. Signing the application and reinstalling should resolve this issue. At this time, Motorola UIQ Platform Security policy requires that all native Symbian applications be signed in order to install to the device regardless of whether or not the application uses a specific capability or not. This message also appears when the date and time of the device is set to a date/time outside of the certificate's valid time frame. Ensure the date and time of the device is set properly.

- 2 "Cannot install" Constraints imposed by a developer mode certificate have been exceeded.

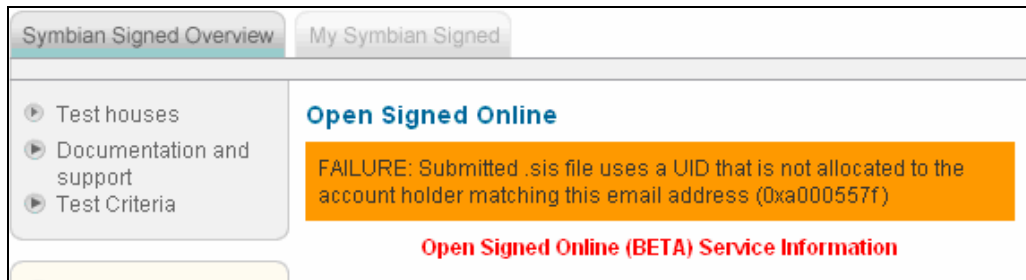
The certificate used to sign this application is not associated to the device IMEI to which the application is being installed. Installing the application to the correct device or resigning with the correctly associated certificates resolves this issue.

- 3 "Cannot install" An upgrade failed because the package being installed is not a valid upgrade.

This message appears when trying to update a signed application (already installed on the phone) using a self-signed application. The way to resolve this issue is to uninstall the signed application and

than install the self-signed application, or, sign the self-signed application and perform the update on the phone.

4 Symbian Signed issues:



This error indicates that the UID used in the application does not match a UID associated with the email address used when signing an application via Symbian Signed online option. Ensure the email address used when requesting Symbian Signed Online signing matches the email address used when requesting UIDs from the Protected range.

General signing guidelines

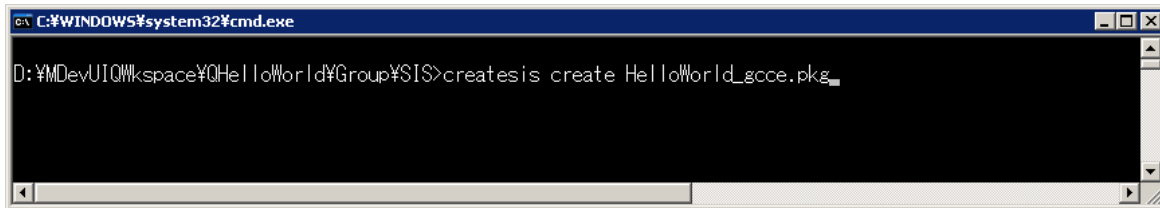
Size limitation on files submitted for signing.

- Open Signed Offline: No effective limit, done locally on your development machine
- Open Signed Online: 4Mb
- Express Signed: 4Mb
- Certified Signed: 40Mb

Appendix C: Self-Signing

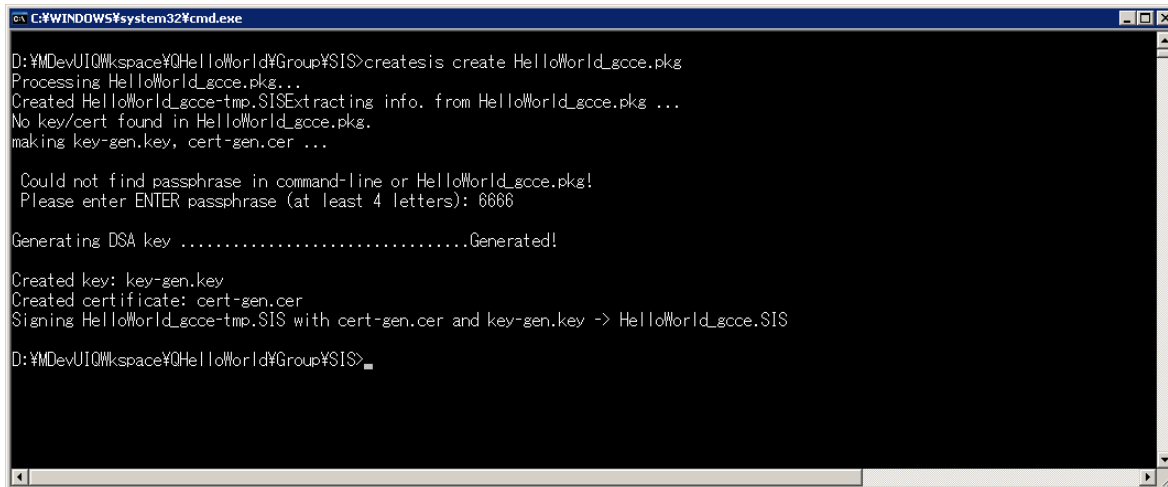
Applications can be signed using the `createsis` command which can take a package `.PKG` file as input and generate a self-signed `.SIS` file as output which can be installed on a Motorola UIQ device. Currently released Motorola UIQ devices allow installation of self-signed applications.

```
>createsis create <package_file.pkg >
```



```
C:\WINDOWS\system32\cmd.exe
D:\MDevUIQ\workspace\HelloWorld\Group\SIS>createsis create HelloWorld_gcce.pkg_
```

When using the `createsis` tool, you will be prompted to enter a four letter 'passphrase' to complete the process. See the UIQ SDK documentation for complete details.



```
C:\WINDOWS\system32\cmd.exe
D:\MDevUIQ\workspace\HelloWorld\Group\SIS>createsis create HelloWorld_gcce.pkg
Processing HelloWorld_gcce.pkg...
Created HelloWorld_gcce-tmp.SISExtracting info. from HelloWorld_gcce.pkg ...
No key/cert found in HelloWorld_gcce.pkg.
making key-gen.key, cert-gen.cer ...

Could not find passphrase in command-line or HelloWorld_gcce.pkg!
Please enter ENTER passphrase (at least 4 letters): 6666

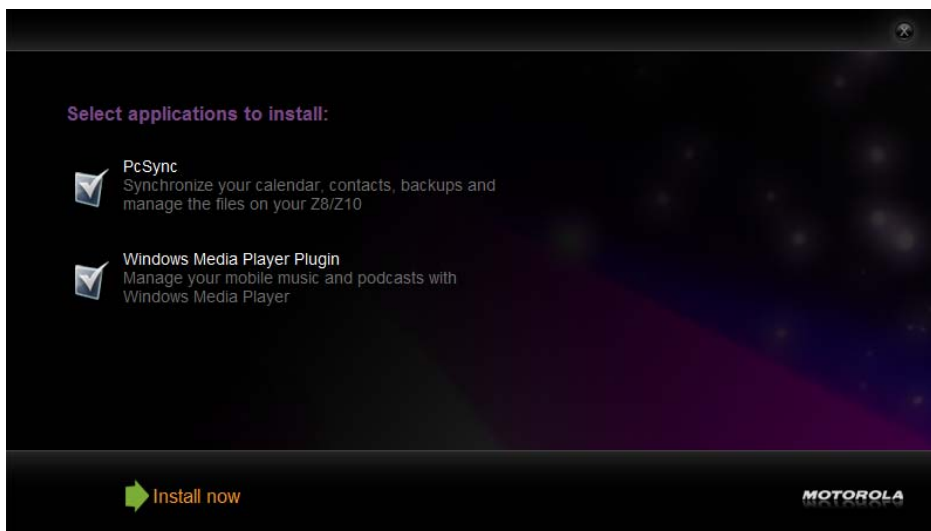
Generating DSA key .....Generated!

Created key: key-gen.key
Created certificate: cert-gen.cer
Signing HelloWorld_gcce-tmp.SIS with cert-gen.cer and key-gen.key -> HelloWorld_gcce.SIS
D:\MDevUIQ\workspace\HelloWorld\Group\SIS>
```

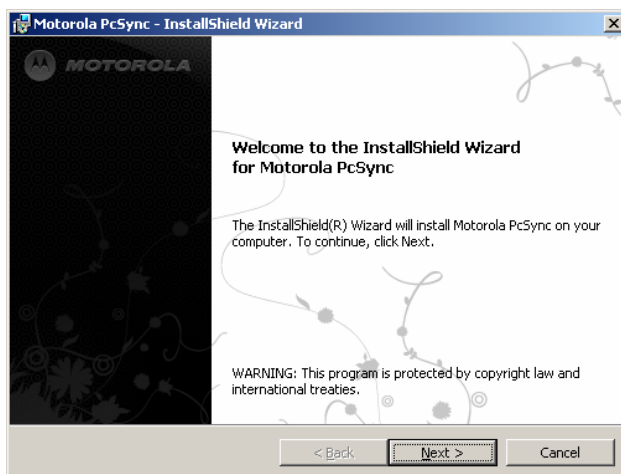
Appendix D: PcSync installation and application deployment

PcSync is available free from the MOTODEV Studio for UIQ download website under Basic Requirements: <https://developer.motorola.com/docstools/motodevstudio/uiq/downloads/>.

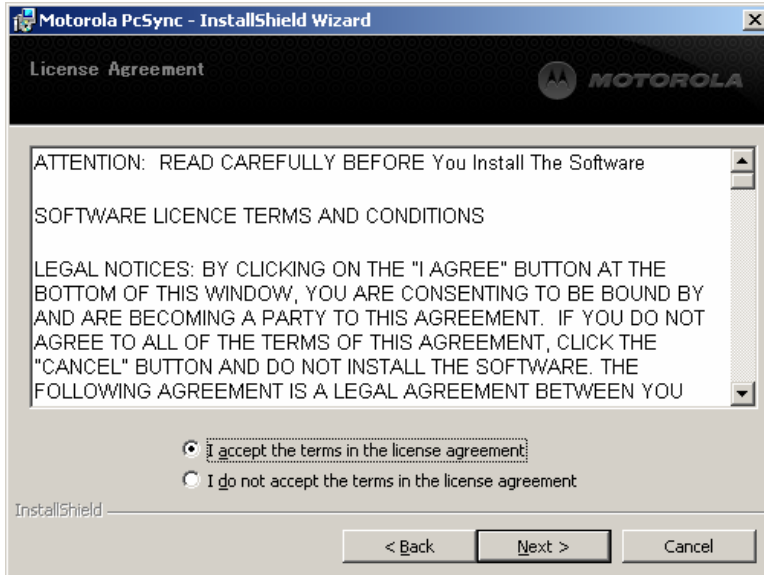
Download the zip file to your PC and unzip the package. Start the download by double-clicking the CD.exe file at the root of the PCSync directory. Select [Install now] button from the Installation Wizard.



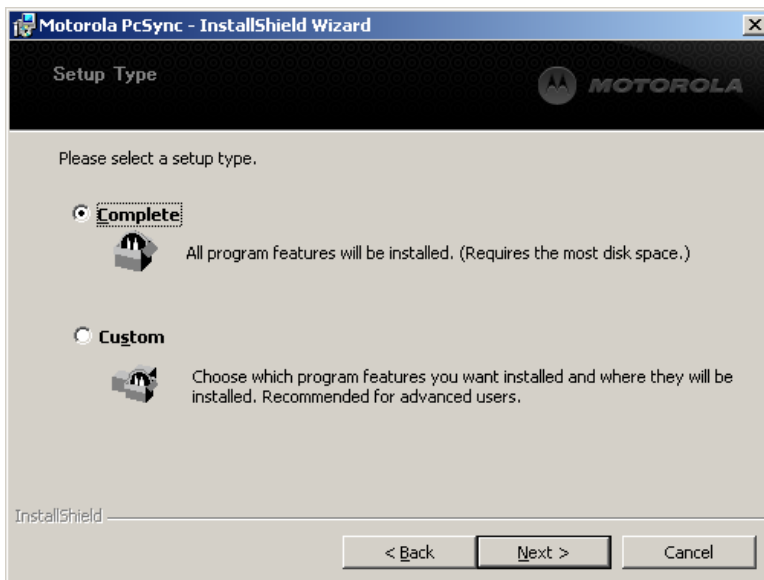
Follow the Installation Wizard steps



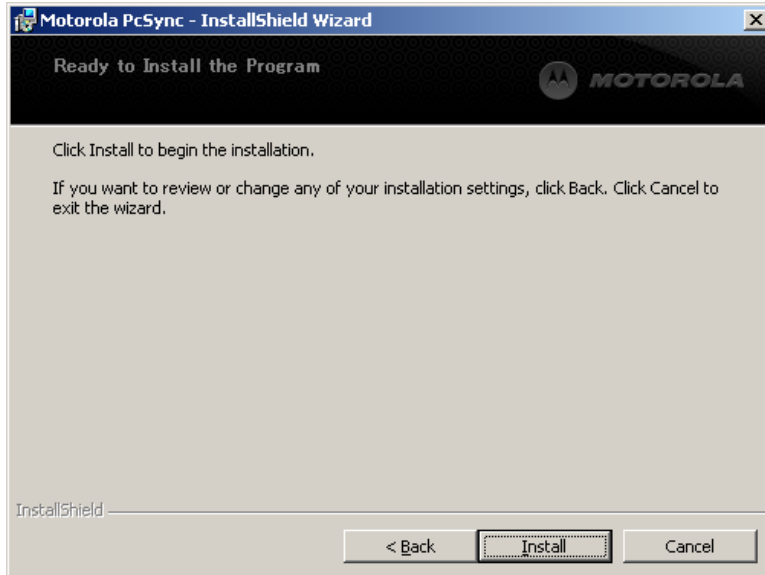
Read the License Agreement then select top radio button (I accept...) to continue installation.



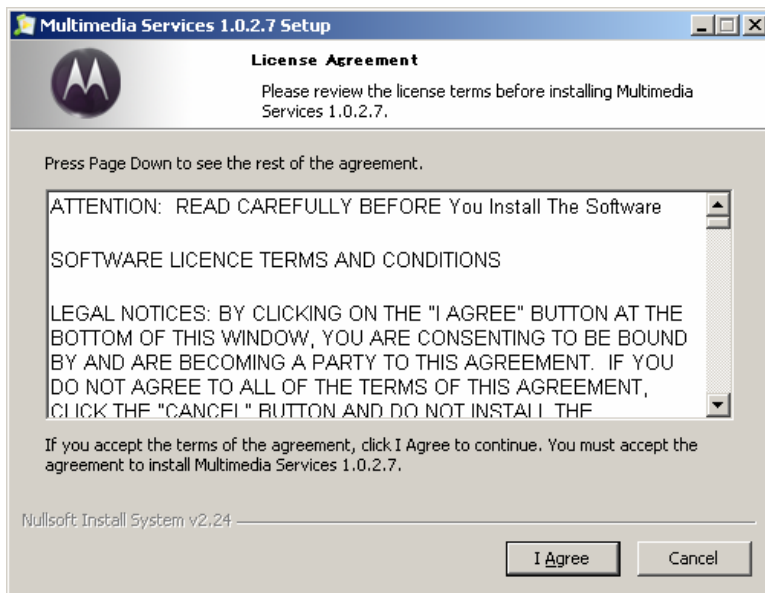
Select [Next]



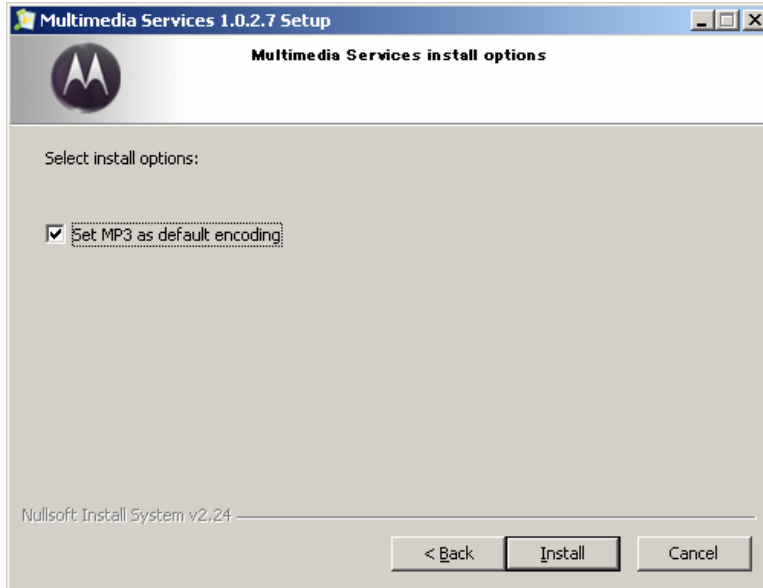
Select [Install]. Allow several minutes for the installation to complete.



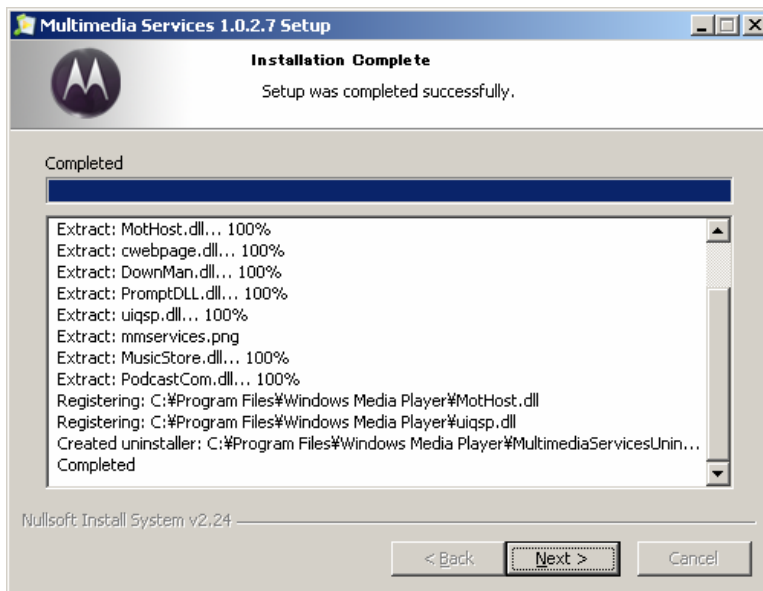
Read and select [I Agree] to continue to install Multimedia Services (Windows Media Player Plugin)



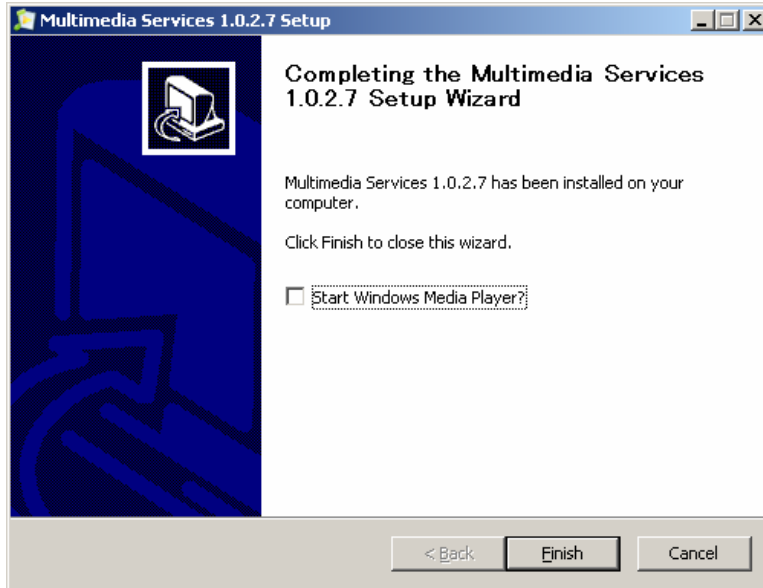
Select [Install]



Select [Next]

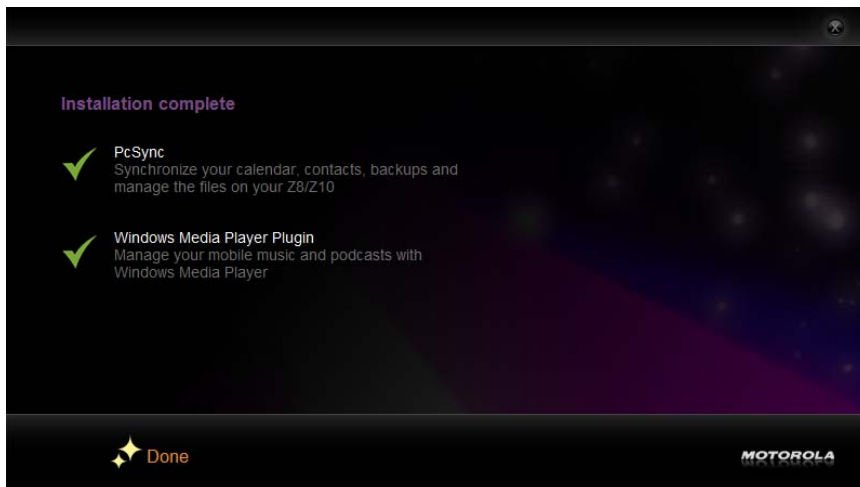


Select [Finish] to complete

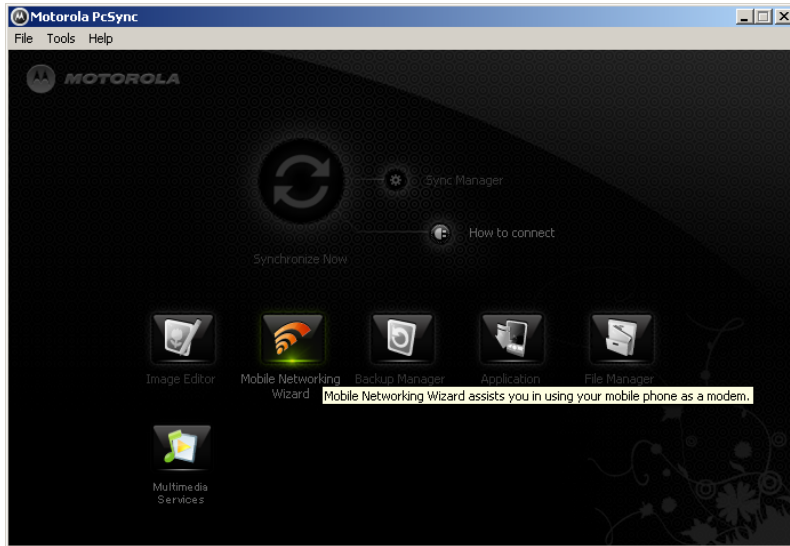


Close Installation Wizard by selecting [Done].

NOTE: Installation completion requires a system restart.



PcSync starts automatically the next time the PC starts. With a USB cable, connect to the MOTO Z8 and select the “*How to connect*” button from PcSync startup screen.



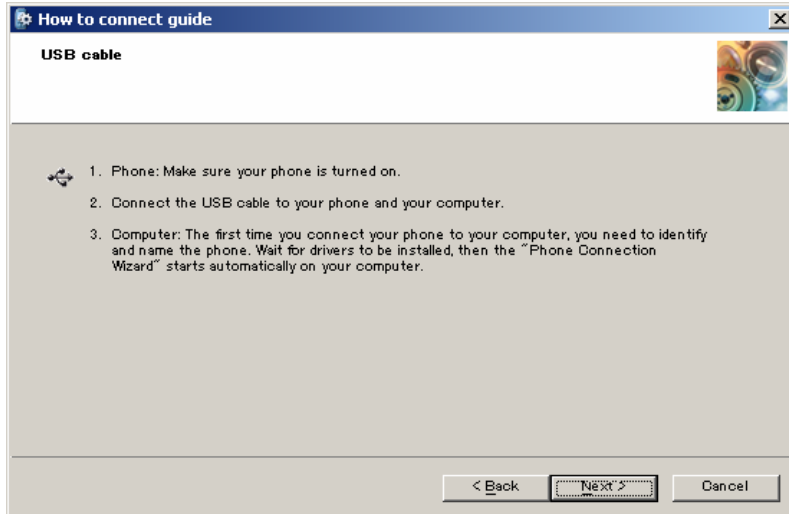
Select [Next].



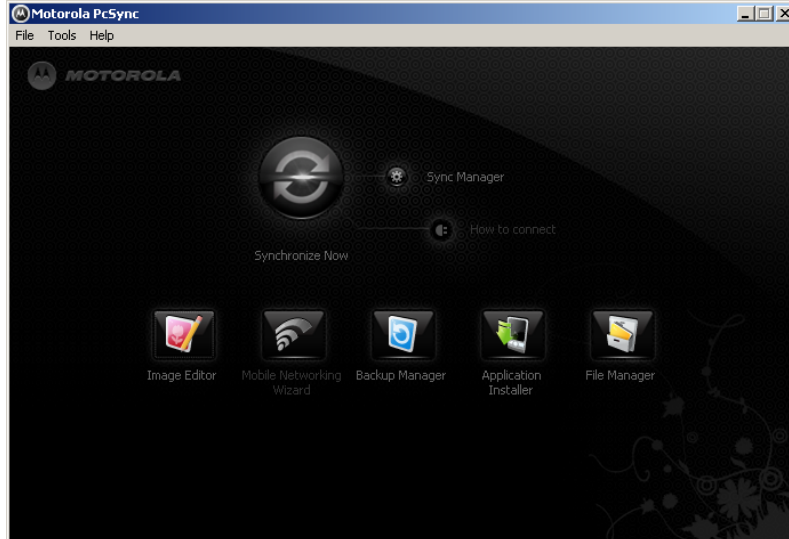
Selec [Next]



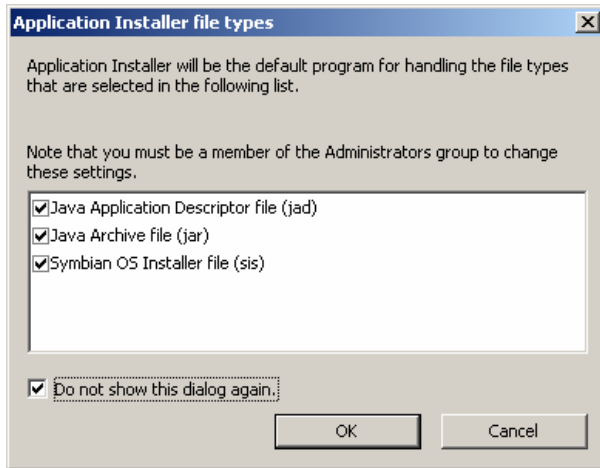
Follow the on-screen directions and select [Next]



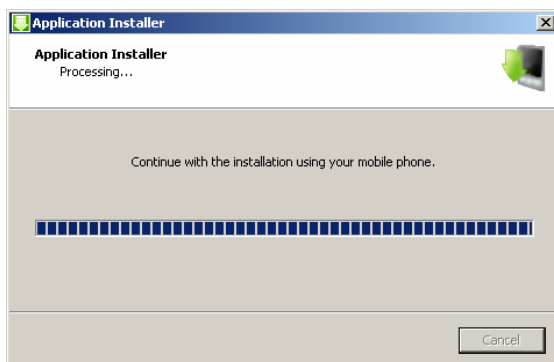
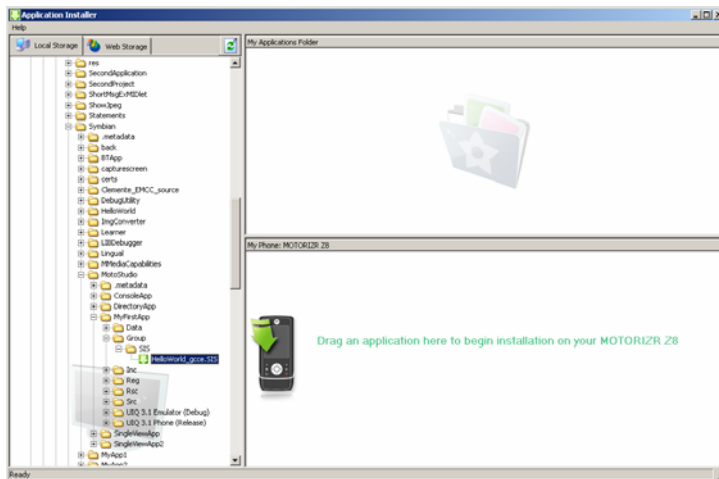
Install the Hello World application to on the MOTO Z8 handset using a USB connection and the PcSync software. Launch PcSync software on your PC from the [Start] > [Programs] > [Motorola] > [PcSync] menu. PcSync software will detect your handset after it is successfully connected via USB and enable the Application Installer button on the main screen.



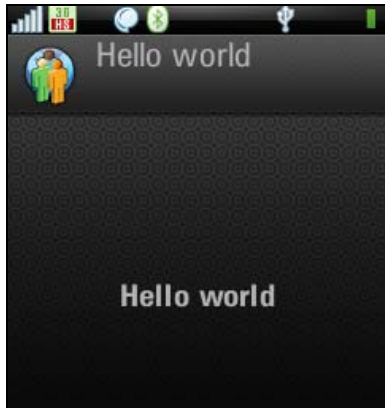
The first time the Application Installer is run you will be prompted with the Application Installer file types window prompt. You may wish to select the [Do not show this dialog again] checkbox to accept the default settings when installing applications.



PcSync software allows you to drag and drop the SIS file previously generated onto the currently connected handset pane.



The rest of the installation process occurs on the target handset itself. Select [Install] > [Continue] (This installs on Phone default location) – [Yes] to run the application. The result is the Hello World application executing on the MOTO Z8 handset.



Finally you may close the PCSync software once the application is successfully installed.

