Porting symbian apps to windows

This article is for S60 2nd Edition or UIQ Symbian C++ developers who want to port their applications onto Windows Mobile 2003-based Smartphones.

Compatability

- S60 SDK 2.1 for Symbian OS, which supports Microsoft Visual C++ .NET
- Microsoft Visual Studio .NET 2003
- Microsoft .NET Compact Framework version 1.0 Service Pack 2 or later
- Microsoft Visual C++ version 4.0 Service Pack 3 or later
- Microsoft Smartphone 2003 SDK Microsoft Visual C#

Porting or Reimplementation?

Many differences exist between the application architecture of a Symbian OS C++ GUI application and the architecture of a similar application based on Windows Mobile platform. These differences are illustrated by a look at that old favorite, the "Hello World" application.

When Is Porting an Option?

Windows Mobile offers application programming interfaces (APIs), such as sockets servers, file system APIs, and database servers, which offer at least the same functionality as equivalent APIs on Symbian OS. Because of this commonality in user input mechanisms and API functionality, you should at least be able to reuse the design in a Windows Mobile version, even if you must put in a lot of work on the reimplementation.

To build the application and run it -

- Click Start, point to All Programs, point to Microsoft Visual Studio .NET 2003, point to Visual Studio .NET Tools, and then click Visual Studio .NET 2003 Command Prompt. A command window opens.
- Change the directory to the \SymbianMileage\Engine subfolder in the sample code folders.
- Type the following command: makmake mileagemanager.mmp vc7This command creates a Visual Studio .NET 2003 solution and project file.
- On the Command Prompt window, change the directory to ..\group.
- Type the following command: makmake mileagegui.mmp vc7 This command creates a Visual Studio .NET 2003 solution for the GUI application.
- Open this solution in Visual Studio .NET 2003, and then click Build Solution on the Build menu to create the MileageGui.app application.
- On the Debug menu, click Start to run the application in the S60 emulator. When prompted to enter the Executable for Debugging Session, type drive where installed:\Symbian\7.0s\Series60_v21\Epoc32\release\wins\udeb\epoc.exe.
- Leave the URL where the project can be accessed box empty, and then click OK.The S60 emulator starts. Navigate to the mileageGui icon in the applications list, and then click OK on the emulator to start the application.

To create the project -

- Open eMbedded Visual C++ 4.0. On the File menu, click New.In the New dialog box, select the Projects tab, and click WCE Dynamic-Link Library.
- Type a suitable name in the Project Name box, such as MileageEngine.
- In the Location box, enter the name of a folder where you want to create the project, such as C:\MSSmartphoneEngine.
- Click OK.In the WCE Dynamic-Link Library – Step 1 of 1 wizard page select A DLL that exports some symbols, and then click Finish.
- Click OK.The New Project wizard generates the code for a DLL that exports symbols.

To copy and modify the files -

- Using Windows Explorer, copy the files mileagemanager.cpp and mileage.cpp from the Symbian version of the application (they are in the \SymbianMileage\engine folder), and then paste them into the C:\MSSmartphoneEngine\MileageEngine folder.
- From the \SymbianMileage\inc folder in the Symbian version of the application, copy the files mileagemanager.h and mileage.h to the C:\MSSmartphoneEngine\MileageEngine folder.
- In eMbedded Visual C++, in the Workspace window, click the FileView tab, and then right-click the Source Files folder to display a menu of options.
- On the shortcut menu, click Add Files To Folder.In the Insert Files into Project dialog box, select mileagemanager.cpp and mileage.cpp, then click OK.
- Add the files mileagemanager.h and mileage.h to the Header Files folder in the same way.

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.